A submission presented in partial fulfilment of the requirements of the University of Glamorgan/Prifysgol Morgannwg for the degree of Doctor of Philosophy

THREE DIMENSIONAL BUILDING RECONSTRUCTION FROM RAW LIDAR DATA

Rebecca (Oi Chi) TSE

April 2008

Certificate of Research

This is to certify that, except where specific reference is made, the work described in this thesis is the result of the candidate. Neither this thesis, nor any part of it, has been presented, or is currently submitted, in candidature for any degree at any other University.

Candidate: ----------

Director of Studies: ----------

Date: ----------

Abstract

Airborne Laser Scanning is an advanced surveying technology (also called Light Detection and Ranging - LIDAR) which mounts a laser scanner on an aircraft. The aircraft scans the Earth's surface and captures data by emitting and receiving light pulses transmitted onto the terrain objects. The captured data are in three dimensions (3D); however no extra information is provided to describe them. Additional algorithms are needed to extract meaningful and useful information from the data.

The popularity of LIDAR has attracted attention as researchers try to develop algorithms for 3D building reconstruction in Geographical Information Systems. The limited information provided by the data makes building boundary and roof structure extractions become essential tasks when analysing the data.

This research examines the limitations of different algorithms for extracting building outlines and remodelling roof structures from the LIDAR data solely, and suggests an alternative approach for reconstructing buildings using raw LIDAR data. Most of the current methods use additional data sources (e.g. cadastral data, aerial photos, or satellite images) and pre-defined building models to reconstruct 3D buildings.

The extraction method proposed starts by re-sampling the captured data in a lower resolution index layer and the aim is to search for vertical wall segments which separate the high and low areas. The wall segments found are connected and modified to form closed building outlines and corners. The roof remodelling system suggested starts by creating a triangulation using the extracted data points which are inside the building boundaries. Three clustering methods are used to separate the triangles into groups which share the same properties (e.g. orientation and geographical location). Each group of triangles represents a plane on the roof. Plane to plane relationships are found, and the building corners and roof ridges are calculated by using the three planes intersection. Finally the building is reconstructed from the terrain model using a set of well-developed toolkits to extend the TIN model with preserved topological connectivity.

Real LIDAR data are used to evaluate the capability and the validity of the developed algorithms. The data were captured in Bournemouth by the Ordnance Survey UK. In conclusion, several suggestions are made to improve the algorithms for future development.

List of Publications

1. Tse, O.C., and C. Gold (2001) "Terrain, Dinosaurs and Cadastres - Options for Three-Dimensional Modelling", Proceedings, International Workshop on "3D Cadastres", Ed by: Christ Lemmen and Peter van Oosterom, Delft, The Netherlands, November, p. 243-257.
2. Tse, O.C., and C. Gold (2002) "Integration of Terrain Models and Built-Up Structures using CAD-Type Euler Operators" Proceedings, International Workshop On Visualization and Animation Of Landscape, Kunming, China, February.
3. Tse, O.C., and C. Gold (2002) "TIN Meets CAD - Extending the TIN Concept in GIS", Proceedings: ICCS 2002 Lecture Notes of Computer Science, vol 2331, Amsterdam, The Netherlands, p. 135-143.
4. Tse, O.C., and C. Gold (2002) "A Surface-Representation Approach to a Three-Dimensional Cadastre" The GIM International Journal, Vol 16, June, 2002 , p. 46-49.
5. Gold, C.M., Dakowicz, M. and Tse, O.C.R. (2002), Visualization and Decision Support for Watershed Management, In Proceedings of ISPRS Commission II Symposium on Integrated Systems for Spatial Data Production, Custodian and Decision Support, Xian, China p. 125-130.
6. Tse, O.C., and C. Gold (2003) "A proposed topological model for a three-dimensional cadastre." Computer Environment and Urban Systems theme issue on '3D Cadastres', July Vol 27 (4), Eds by: Christ Lemmen and Peter van Oosterom, July, 2003, p. 427-445.
7. Tse, O.C., (2003) "3D Terrain Models". Dagstuhl Seminar No 03401, Computational Cartography and Spatial Modelling. Organizers: P. van Oosterom (Delft University of Technology, NL), M. Sester (Univ. Hannover, D), J. Snoeyink (UNC Chapel Hill, USA), M. Worboys (Univ. of Maine at Orono, USA), 28 Sept - 3 Oct, 2003, IBFI Schloss Dagstuhl, http://www.dagstuhl.de/03401/Proceedings/.
8. Tse, O.C., and C. Gold (2004) "TIN Meets CAD - Extending the TIN Concept in GIS", Future Generation Computer Systems (Geocomputation) Vol.20, No. 7, Eds by: Yong Xue, Xiangyu Sheng, Narayana Jayaram, Oct, 2004. p. 1171-1184.
9. Tse, O.C., C. Gold and D. Kidner (2004) "An Original Way of Building a TIN with Complex Urban Structures" Proceedings, International Society of Photogrammetry and Remote Sensing XXth Congress, 12-23 July, 2004, Istanbul, Turkey.

10. Tse, O.C. and C. Gold (2004) "3D Urban Terrain Modelling with Bridges and Holes" Proceedings, The 30th Congress of the International Geographical Union, August, 2004, Glasgow, UK, p. 15-20.

11. Tse, R. O.C., M. Dakowicz, C. Gold and D. Kidner (2005) "Automatic Building Extrusion from a TIN model Using LiDAR and Ordnance Survey Landline Data" Proceedings, Geographical Information System Research UK, extended abstract, April, 2005, Glasgow, UK, p. 256-264.

12. Tse, R. O.C., M. Dakowicz, C. Gold and D. Kidner (2005) "Building Reconstruction Using LIDAR Data" Proceedings, Dynamic and Multi-dimensional GIS 2005, September, Pontypridd, Wales, UK, p. 156-161.

13. C. Gold and R. O. C. Tse (2005) "Quad-edges and Euler operators for automatic building extrusion using LiDAR data" Seminar Geo-information and Computational Geometry, Utrecht University, The Netherlands, November 2005.

14. Tse, R. O.C., C. Gold and D. Kidner (2006) "A New Approach to Urban Modelling Based on LIDAR" Proceedings: Winter School of Computer Sciences 2006, the 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2006, Plzen, Czech Republic, p. 279-286.

15. Tse, R. O.C., C. Gold and D. Kidner (2007) "3D Building Reconstructions Using Raw LIDAR Data" Oral Presentation in the SET for Britain event at the House of Common (Short-listed in the best poster award).

16. Tse, R. O.C., C. Gold and D. Kidner (2007) "Using the Delaunay Triangulation/ Voronoi Diagram to extract Building Information from Raw LIDAR Data" Proceedings: International Symposium on Voronoi Diagrams in Science and Engineering, 2007, Pontypridd, U.K. p.222-229.

17. Tse, R. O.C., C. Gold and D. Kidner (2007) "Building Reconstruction Using LIDAR Data", Proceedings, Dynamic and Multi-dimensional GIS 2007, August, Urumchi, China, pp. 121-126.

18. Tse R. O C., "3D Building Reconstruction with LIDAR Data", Proceedings of the 1st Research Student Workshop, March 2007, Pontypridd, U.K., p.94-98.

19. Tse, R. O.C., C. Gold and D. Kidner (2007) "3D City Modelling from LIDAR Data" , Proceedings, 3D GeoInfo 2007, Delft, the Netherlands, 2007, December, p.161-175.

20. Tse, R. O.C., C. Gold and D. Kidner (2008) "Implementation of Building Reconstruction Algorithm Using Real World LIDAR Data", SDH-SAGEO 2008 Conferences, Montpellier, France, 2008, June, p.297-313.

Acknowledgments

I would like to express my deepest thanks to my lord, Jesus Christ. He prepared the best for me these past five years including my marriage and our one year old daughter. He walked with me throughout the whole period of my research. His saving grace is more than enough because he said, "My grace is all you need (II Corinthians 12:9)".

I met Prof. Chris Gold in Hong Kong and he has supervised my study since 2000. His enthusiasm for research awakened my research interest and encouraged me to start my second research degree in Wales. I would like to send my sincere gratitude to him that he has guided, supported and encouraged me for the last eight years.

I also would like to thank Dr. Dave Kidner. Without his help to look for funding for my project, it would have been difficult for me to finish my studies in the UK. Thank you also for giving me valuable advice for conducting the research.

I would like to thank Mrs. Valerie Gold for helping proof read the thesis and giving me valuable advice in writing it. Special thanks go to the members of "The Voronoi team", for their valuable contributions to my research. Maciej Dakowicz gave me really useful advice in developing the program. Hugo Ledoux gave me a big hand while writing the thesis.

Many thanks go to all my office mates in Wales (Marylin, Fatma, Michael, Paul, Andrew and others) that made my time more enjoyable, and also to my friends (Pauline, Christine, Yeung and others) who made me welcome though I was studying far away from home.

I would like to send my special thanks to the Ordnance Survey, United Kingdom. The research discussed and the data used in this research was funded, provided and supported by the O.S. (Research and Innovation).

Last but not least, my special thanks go to my family, particularly my husband Fred and my little princess Tania. They have given me the greatest support ever to finish my research. Without their help, my research would have been much more difficult. Thank you to all the others who also helped me with the dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Maps have changed dramatically over the last 40 years. They are represented not only on paper, but also in digital format (computer mapping). Nowadays, more information is required, for example the third (elevation) and even fourth dimensions (time). Luckily, these can largely be fulfilled by today's advanced computers. Geographical information systems (GIS) help to computerize map making geographical data analysis to improve the quality of maps.

But what is GIS? There are many different definitions of GIS, although many are similar. GIS can be defined as an information system to input, store, retrieve, manipulate, analyse and output geographically referenced or geospatial data. Nowadays most commercial GIS packages can handle and display two and a half dimensional (2.5D) or fully three dimensional (3D) data easily.

An increasing need for automated 3D city reconstruction models is being driven by a variety of applications, from tourism to disaster management. The three main issues of the reconstruction are: display technology, data structures and data collection.

Display technology is now satisfactory even on modest computers, no matter what hardware and software. The economical, fast and advanced display card technologies provide a really good support for 3D visualisation on computers. The Games industries have developed many advanced and free graphic libraries (software) which give an excellent environment to develop 3D modelling.

Data structures connecting the various portions of buildings and terrain have been well defined in Computer Aided Design (CAD), computational geometry and GIS. Tse (2003) tried to integrate them to extrude buildings with vertical walls and create tunnels or bridges from the Earth's surface while the topological connectivity was preserved for further spatial development. Tse and Gold (2003) developed a set of simple and user friendly toolkits for 3D building reconstruction.

Modern technologies provide many ways to capture information from the Earth's surface, for example: traditional surveying, photogrammetry, Global Positioning Systems (GPS) and Light Detection and Ranging (LIDAR). Traditional surveying and GPS surveying methods capture

data point-by-point which is generally time-consuming. Photogrammetry and LIDAR can capture a huge amount of data in each survey, but only LIDAR can capture data day or night.

The arrival of LIDAR - airborne laser altimetry - now provides rapid elevation models of terrain and buildings. One of the greatest remaining difficulties is to extract building information efficiently from the captured data and to convert this information into CAD-type models containing walls, roof planes and terrain which can be rapidly displayed from any 3D viewpoint.

Airborne laser survey captures man-made (buildings) and natural (trees and forest) objects on the earth's surface at the same time. It is common in current research to remove all buildings, trees and terrain objects to generate a so-called bare-earth model. Then building boundaries are obtained by searching, with the help of additional data sources or existing 2D map data (Sohn and Dowman, 2003, 2004; Suveg and Vosselman, 2004, 2001; Vosselman and Dijkman, 2001). Pre-defined building models are chosen which fit the building outline. Finally, CAD is used to create the building and paste it on top of the bare-earth model.

Problems occur with their approach when no other data source is available for building outline extraction, when the target building does not match any pre-defined building shape, and if topological connectivity between the building and the terrain surface is needed for further spatial analysis.

1.1 LIDAR data and Building Information Extraction

Airborne LIDAR surveying is a relatively new technique for capturing data from the Earth's surface. This is faster, denser and more efficient than traditional land surveys and photogrammetry (Fowler, 2001; Flood, 2001).

LIDAR survey, also called airborne laser scanning, is a laser scanner mounted in an aircraft to scan across the Earth's surface. A time measurement is taken when the laser beams are emitted and reflected back to the scanner. With the help of the inertial navigation systems (INS) and global positioning systems (GPS), the height and coordinates of the scanned area are calculated. The details of LIDAR surveying is shown in Chapter 2.

An example of a last-return LIDAR dataset collected at Bournemouth in the UK (from the UK Ordnance Survey) is shown in Fig. 1.1, where 1.1(a) and 1.1(b) show the 2D and 3D views respectively. The amount of information contained in such high-density 3D point clouds is enormous. The shape of the buildings and the roof structures are all easily identified in Fig. 1.1(c). The human eye can find the building information in the 3D view quite easily, but not in 2D. Undoubtedly the computer will not be able to perform any better than human eyes at capturing to capture the information based on visual interpretations only. Extracting building boundaries using LIDAR data becomes an attractive and interesting challenge.

Brunn and Weidner (1997b); Bonn (1999); Gerke et al. (2001); Mikhail (2000); Suveg and Vosselman (2004); Henricsson and Baltsavias (1997) have all developed different approaches to extract and reconstruct buildings from aerial images, which is quite similar to extracting information from LIDAR data. In current research, it is common to combine different data sources to reconstruct buildings from LIDAR data. Some of the research combines LIDAR with satellite data (Sohn and Dowman, 2001, 2004), or aerial images (Rottensteiner, 2001; Rottensteiner and Briese, 2002; Khoshelham, 2005) during the reconstructions.

LIDAR data contains not only the building footprint, but also the roof structures of the buildings. Many buildings have complicated roof structures and it is not easy to discover the roof styles from the x, y and z coordinates only. Some research uses a model-driven approach to search for a best building model to fit the data points (Brenner and Haala, 1998; Haala and Brenner, 1999; Hofmann et al., 2003; Rottensteiner and Briese, 2003). Other research uses a data-driven approach to reconstruct buildings, but they require high-resolution LIDAR data during the building reconstruction (Vosselman, 1999a; Rottensteiner and Briese, 2003).

Different methods have been developed to solve the problem; however there is still room for improvement. The aim of this research is to look for an alternate solution to reconstruct 3D building models without using additional data source, pre-defined models, and high resolution data, and with preserved topological connectivity.

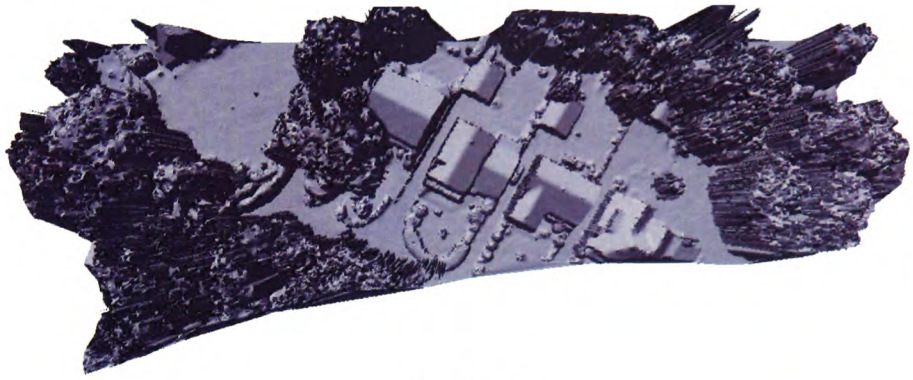
1.2 3D Buildings Modelling Using Triangulation

A common way to construct a 3D building model on a terrain surface is to paste the rendered 3D buildings on top of a grid-based terrain surface. It is a simple way to reconstruct the 3D building models; however no topological connectivity is preserved during the construction procedures. Buildings pasted on the terrain look like playing "Lego^{TM}".

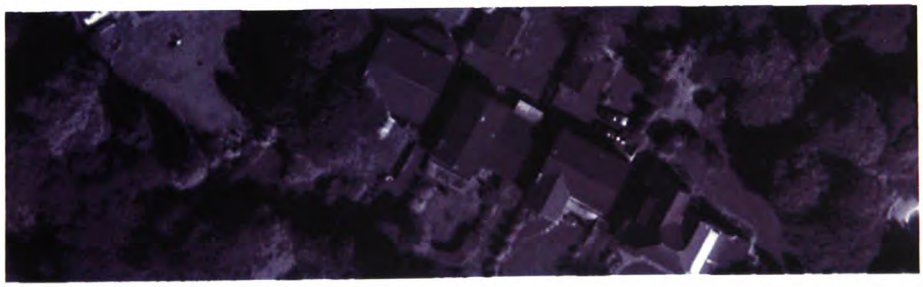
Airborne laser scanning sweeps across the ground surface to capture man-made and natural objects, but it returns an irregular point pattern cloud. Many current algorithms filter the captured data and generate a bare-earth model (Jacobsen and Lohmann, 2003; Elmqvist, 2001; Roggero, 2002; Vosselman, 2000; Brovelli et al., 2002). This research extracts and reconstructs building models based on raw LIDAR data only; therefore no filtering algorithm is used. Using raw LIDAR data can avoid data being lost during the processes of filtering and interpolation.

Within the GIS community the raster structure (grid) or triangulated irregular networks (triangulation) are two different ways to model the elevation of a terrain. In 2D, a raster structure is a regular tessellation with grid squares (Fig 1.2(a)), and to each element (called a pixel) is assigned the value of an attribute (e.g. height or time). The simplicity of the structure makes it easy to manage with the computer because data are stored row by row and column by column. The location of an object is represented by the sequential position of pixels in an array. However

(a) 2D view

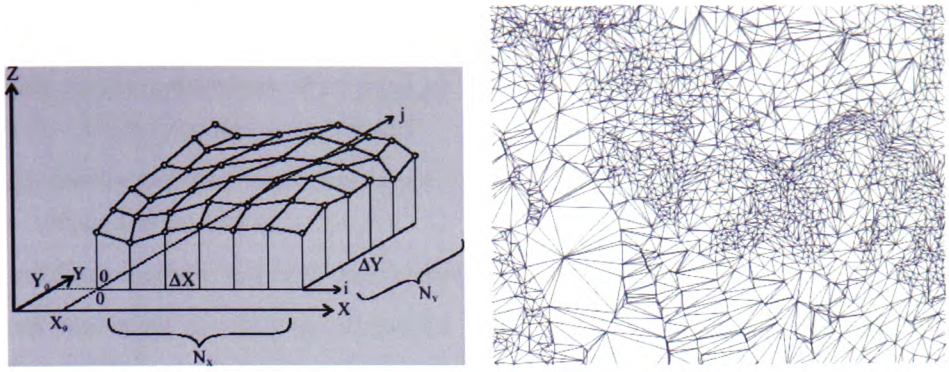


(b) 3D view



(c) Aerial Image

Figure 1.1: A sample LIDAR dataset (from the UK Ordnance Survey)



(a) Raster Structure (b) Triangulation

Figure 1.2: The Modelling System

using more data points to represent an important area may not be possible and the fixed grid size makes it difficult to split a cell into smaller cells. Some of the disadvantages of the raster structure make it unsuitable for modelling the terrain surface (Kemp, 1993; Haklay, 2004). On the other hand, triangulation seems more flexible in Fig. 1.2(b). It is an irregular tessellation, filling up the terrain with triangles which are formed by the x and y coordinates. The elevations of the triangle vertices uplift the triangles to describe the topography of the area. Although the complexity of storing, creating and analysing triangulation is higher than for a grid, the flexibility of using triangulation seems to be attractive for terrain modelling.

A flexible data structure is needed for modelling the terrain in this research; therefore triangulation is used as the terrain modelling data structure. At the same time, my previous research provides a solid foundation for modelling the 3D buildings with guaranteed topological connectivity (Tse, 2003). The model starts with a 2.5D triangulation and is extended to include buildings, bridges and tunnels. The Quad-Edge is used as the basic data structure in this research, which is a completely object-oriented structure. Gold (1998) has implemented the data structure using object-oriented programming methods. Delphi is the object-oriented programming language to be used in this research.

1.3 Objective of the Research

The objective of this research is to create an algorithm with guaranteed topological relationships to reconstruct buildings automatically from LIDAR data, but no pre-defined building models and no additional data sources are required during the process. The topological connectivity of the reconstructed model is preserved for further spatial analysis. The approaches are based on the Delaunay triangulation and its dual, the Voronoi Diagram, because the flexibility of the triangulation and Voronoi cells allows the cells to be split into different sizes.

The following propositions are made to achieve the objective of the research:

1. Buildings are collections of contiguous elevations that are higher than the surrounding terrain. Their boundaries are "walls".
2. Walls have a specified minimum height, and this height difference is achieved within a very few "pixels (data points)".
3. A building consists of a high region entirely surrounded by walls.
4. Roofs are made up of planar segments, most of whose constituent triangles have similar vector normals, orientation and geographical location.
5. The advanced relationships between roof planes may be represented as a dual triangulation.

1.4 Scope of the Research

This research aims at developing a prototype GIS that manages to extract and reconstruct 3D buildings from raw LIDAR data only. The research is conducted from an algorithmic point of view; its main results are a set of algorithms for extracting and reconstructing 3D buildings, and the processing of real data using the developed algorithms. The design of the algorithms allows the user to reconstruct 3D buildings using raw LIDAR data without any additional data sources, and no pre-defined building models.

The first step is to search for building outlines by separating the high from the low areas. However if a building is constructed on a slope or partly under the Earth's surface, it may not be possible to detect it. The height difference between the high and the low land must be larger than a certain value (for example 2.5 meters). If not, further research is required to find the vertical wall segment.

The stages of the reconstruction are given below:

1. Extract building boundaries.
2. Create a triangulation using points (on the roof) inside the building outlines.
3. Cluster the roof triangles with three clustering algorithms.
   a Orientation Clustering.
   b Perpendicular to Orientation Clustering.
   c Geographical Clustering.
4. Create roof planes for each group of triangles.
5. Find the relationships between the roof planes.
6. Calculate the building corners and ridges using the plane to plane intersection.
7. Reconstruct buildings from the terrain surface.

The goal of this research is not to design a perfect commercial product to handle huge amounts of data, but rather a simple system to implement and verify the algorithms. However it is a prototype system, only one building is built at a time. The approaches of reconstructing more than one building are the same as reconstructing only one building. On the other hand, the reconstruction procedures are semi-automated. The input of the system include the raw LIDAR data and some tolerances, and the output is the reconstructed building on the Earth's surface. The system can be improved to fully-automated in the future.

The following topics are closely related to the 3D building reconstruction from LIDAR data that are not covered in this research:

- Filtering algorithms: in this research no filtering algorithm is developed because no bare-earth model is needed for the reconstruction.
- Very large datasets: this research is concerned with the development of algorithms rather than on end product. A small amount of data is used to verify the algorithms.
- Data quality and errors: all input LIDAR data are assumed to be 'noise-free' and 'error-free'.

1.5 Outline of the Thesis

The thesis is organised in eight chapters (including this one). Chapters 2 and 3 carry a literature review for data collection, filtering and building extraction algorithms, and methods of building reconstruction. Chapters 4, 5 and 6 show the algorithms and procedures of the building reconstruction. Some synthetic data were used to illustrate the ideas and algorithms in the chapters. Chapter 7 evaluates the proposed algorithms using real data.

Chapter 2 introduces different surveying methods. Different surveying methods are reviewed which include GPS, photogrammetry and airborne laser scanning. A part of the chapter concentrates on several ways to capture data points and it explains the reason for using LIDAR data points.

Chapter 3 shows some existing research approaches to the reconstruction of 3D buildings, providing a literature review of the techniques and backgrounds of the thesis. The chapter focuses on describing filtering, building extraction and reconstruction algorithms of the data points. The chapter concludes by highlighting the shortcomings of available techniques for 3D building models.

Chapter 4 presents the algorithms for extracting building blocks that look for the wall segments to separate high and low areas. The vertical wall segments are found by splitting the Voronoi cells. Three methods are described in the chapter. However the last method is only in a

preliminary testing stage. Although more work is required, it is still considered to be worth mentioning. Artificial data are used to explain the algorithms defined in Chapters 3, 4 and 5.

Chapter 5 shows the algorithms used to remodel roof structures. Roofs are made up of planar segments. All the building corners and ridge points are obtained by intersecting the roof planes. Two methods are shown. The first works for simple gable roofs only, while the second one works for more complicated roof structures.

Chapter 6 discusses the algorithms for extruding and reconstructing buildings. Methods from previous research are used to extrude buildings from the Earth"s surface. A program manual is also provided in this chapter that gives some instructions on how to use the system in this chapter.

Chapter 7 implements the developed algorithms using real LIDAR data and evaluates the results and the algorithms. The real LIDAR data are supplied by the Ordnance Survey in the UK.

Chapter 8 concludes the thesis by summarising the advantages of using the proposed algorithms to reconstruct buildings, and by outlining the major contributions of this research. Recommendations for further research are also given.

Chapter 2

General Background - Why LIDAR?

3D city models are increasingly popular nowadays not only for visualisation purposes, but also for flood modelling, 3D spatial analysis, neighbourhood analysis, and so on. The demand for models includes a strong focus on 3D building reconstruction research. A 3D city model is a 3D computer model of an urban environment which is being driven by a variety of applications, from navigation to city planning systems.

This chapter shows different kind of surveying techniques and isolates the best one for creating 3D city models. A brief introduction to GPS surveying, photogrammetry and airborne laser scanning will be given in the first three sections. Among these advanced surveying techniques, there must be a reason why laser altimetry is continually gaining popularity. Therefore the advantages and disadvantages of each surveying method will be shown. The last section explains the reasons for using laser scanning data in this project.

2.1 Global Positioning Survey

The Global Positioning System is a satellite-based navigation and precise-positioning system which has 24 satellites placed in orbit. It provides three-dimensional locations, velocity and time information to fully equipped users anywhere on or near the surface of the Earth (or sometimes off the Earth).

Originally GPS was developed by the US Department of Defense (DoD) to use in the military area. In 1980, the US government made the system available for civilian use. Today GPS can be used 24 hours a day, anywhere in the world and in any weather conditions without any subscription fees or setup charges (Hofmann-Wellenhof et al., 1992). The development of GPS has been quick, and it has become a common tool in daily life for land surveying, intelligent vehicle location and maritime navigation. The smallest of today's GPS device is wrist watch size.

GPS satellites circle the earth twice per day and transmit signals. To receive this information, a GPS receiver is used to calculate the user's exact location. At least four GPS satellites are needed to locate a user position (latitude, longitude, altitude) (Limited, 1996) by measuring the distance between the satellite to the receiver (the user location).

Section 2.1.1 describes the properties and functions of GPS. The accuracy of using GPS in surveying will be described in the second section. Then the advantages and disadvantages of using GPS are discussed.

2.1.1 Properties of GPS

Space, control and user segments are the three main components in GPS. The following sections will show each segment. The space and control segments are briefly described because this research mainly uses GPS instead of developing it.

The Space Segment The space segment includes the GPS satellites in orbit around the Earth. Twenty-eight satellites in 6 orbital planes provide information from space Fig. 2.1. The satellites broadcast radio frequency (RF) ranging codes and navigation messages.

The Control Segment The main function of the control segment is 'tracking the satellites for the orbit and clock determination and prediction modelling, time synchronisation of the satellites, and uploading the data message to the satellite' (Hofmann-Wellenhof et al., 1992).

The User Segment The user segment includes militarily and publicly used GPS receivers. They receive, understand and process the radio frequencies transmitted by the GPS satellites. As GPS becomes increasingly popular many different manufacturers produce GPS equipment for both military and civilian activities. The main function of using GPS is for navigation in three dimensions. GPS receivers are made mainly for aircrafts, ships, ground traffic and for hand carrying.

2.1.2 Accuracy of GPS

The accuracy of using a GPS to capture data is very good. However six different types of error can occur when using a GPS, they are: satellite, propagation, receiver, earth, algorithmic and stochastic errors. On the 1st May 2000 the United States announced that they would stop the intentional degradation of the GPS signals available to the public. This improved the accuracy for civilian users of GPS to pinpoint locations by ten times (Dana, 2000).

Real Time Kinematics (RTK) methods of GPS with a certain method of observation can collect elevation and coordinate (x, y and z) data to an accuracy of between one millimeter to one

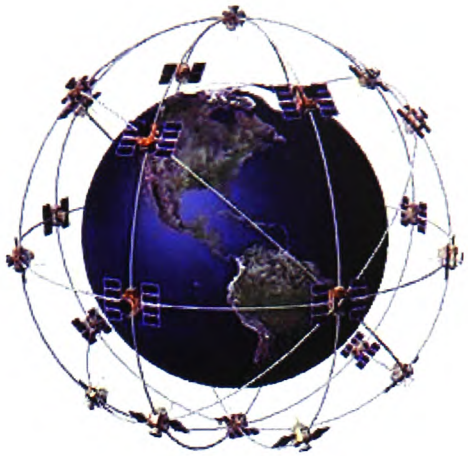


Figure 2.1: 28 satellites in 6 orbital planes

centimeter (Kavanagh and Bird, 2002). Several limitations of using GPS still exist. For example, the signal can be obstructed by high buildings, trees or terrain and result in an inability to capture a position.

The average positioning accuracy of a low cost stand-alone GPS receiver is expected to be 13 meters horizontally and 22 meters vertically, 95% of time. There are several methods to improve the accuracy of using a GPS, e.g. differential GPS (DGPS) or navigation aids to enhance position reliability (map-matching). In this project, we are not interested in improving the accuracy of using a GPS; therefore the improvements are not described.

2.1.3 Advantages and Disadvantages

According to Veneziano et al. (2004) there are similar advantages and disadvantages of using different kinds of GPS e.g. a stand-alone or RTK GPS.

Advantages of Using GPS

- It allows rapid download of data to computer because all of the instruments are in digital format.
- A description/ note on each surveyed point can be taken when doing a GPS point survey.
- The survey procedure works independently which allows rapid data collection.
- One person is enough for a GPS survey, and pre-survey preparation is required.

Disadvantages of Using GPS

- A limited area can be surveyed each time.
- It is difficult to access private areas.
- Sometimes a surveyor needs to go to hazardous areas, for example near motorways or the top mountains.
- Some areas receive ineffective signals which can be blocked by trees or buildings.

GPS is a handy and easily operated instrument which provides incredibly high accuracy data. To create a 3D city model, it needs to capture a large amount of data in a huge area. It is not suitable to use GPS only because it can be time-consuming for measurement.

2.2 Photogrammetric Survey

Photogrammetric survey is an efficient method to capture a huge number of data points every time. So what is photogrammetry? It was defined by the American Society for Photogrammetry and Remote Sensing (ASPRS) in 1956, for the mapping sciences; *photogrammetry is the art, science, and technology of obtaining reliable information about physical objects and the environment through the process of recording, measuring, and interpreting photographic images and patterns of electromagnetic radiant energy and other phenomena.* From the scientific point of view, photogrammetry can also be a mixture of geometry, mathematics and physics.

Photogrammetry can be used for capturing height information. A pair of 2D photos (called a stereopair) with overlapping areas is needed to reconstruct a reliable and accurate stereomodel with height information. A stereopair is taken from two different locations on an aeroplane from a certain height above the surface. The difference in position along the flight line is measured to determine the height of the objects in the photos. Fig. 2.2 shows a stereopair with an overlapping area. With the help of the advanced GPS, the location of the exposure station can be precisely determined in-flight. It requires the relationship between the perspective centre of the camera and the centre of GPS receiver to be measured.

The following sections, discuss the properties, accuracy, advantages and disadvantages of photogrammetric surveying.

2.2.1 Properties of Photogrammetry

Nowadays, processing of photogrammetry has become more automatic. Multi aerial photos (images) in Fig. 2.3 are taken by digital cameras on the plane. The photos are downloaded into the computer for further processing. The overlapping areas are used to produce the stereomodel

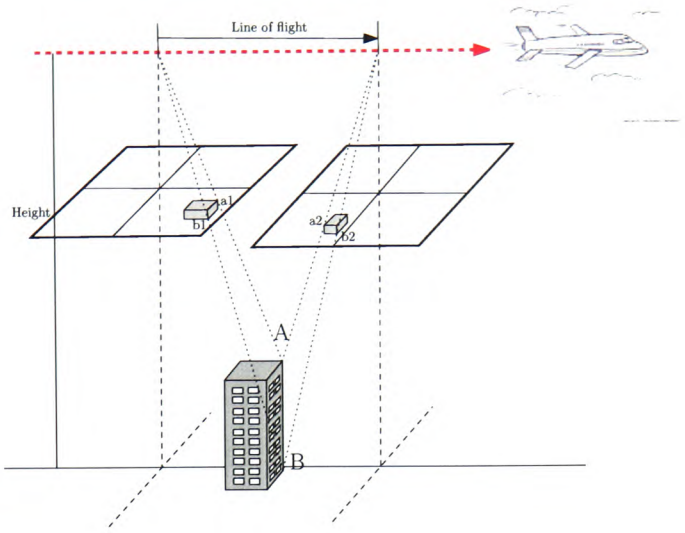


Figure 2.2: A stereopair

with the help of up-to-date hardware and software, and the height of the data points can be extracted from the stereomodel.

Digital techniques are widely available. Photos are not on film but can be recorded and stored on tape or disc. Each pixel from each picture has its known position and measured intensity value. Both black and white or colour or multispectral images are used in the measurements.

Digital photogrammetry uses the same algorithms as analytical photogrammetry. The photo or image coordinates (X, Y, Z coordinates on the photos) are the input and the output are the object's actual positions (with x, y and z coordinates) on the ground. Two steps (exterior and interior orientation) are performed before getting the coordinates of the objects.

Interior Orientation Interior Orientation (inner orientation) is the first step when processing aerial photos. It finds out the line of flight from the photos and rotates both photos according to the flight line. The interior orientation is largely done automatically by software now. Exterior orientation is performed after interior orientation.

Exterior Orientation Exterior orientation includes relative and absolute orientations. Space relative orientation rotates the image's position into the same relationship toward each other as when they were taken. Absolute Orientation chooses some points with known coordinates on the aerial photos and input the coordinates for aerotriangulation.

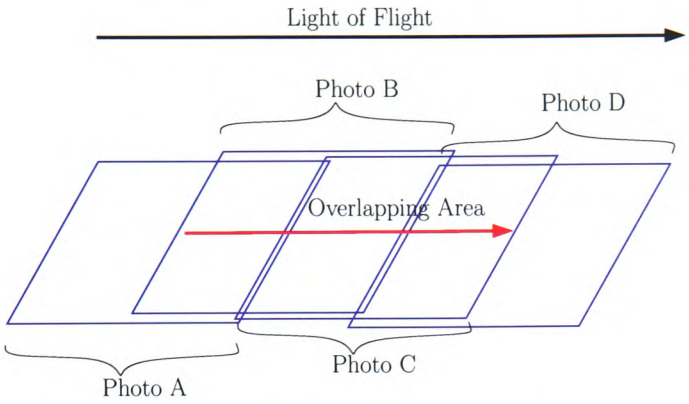


Figure 2.3: A series of aerial photos with an overlapping area.

Exterior and absolute orientations have become more automatic and easy to do with modern software and hardware. The users have to input several points in both of the orientation, the computer can calculate the rest for the user.

2.2.2 Accuracy of Photogrammetry

Photogrammetric surveying is different from traditional surveying because the surveying products are not x, y and z coordinates (or numerical output). It produces a group of photos from which users have to extract information. The accuracy of photogrammetry relies on many different factors. They are:

- higher photo resolution achieves higher accuracy because objects can be measured more precisely. It is affected by the capabilities of your digital camera or film scanner.
- camera calibration, which is a process of determining the camera's focal length, format size, principal point and lens distortion. At present much commercial software packages provide well-developed camera calibration for users.
- photo orientation quality which contributes a lot to the accuracy of the project. It can be improved if the number of well-positioned points increases and the points cover a greater percentage of the photograph.
- photo redundancy, if the same object on the ground appears in as many photos as possible, this is the photo redundancy which helps to correct errors and increase the accuracy.

2.2.3 Advantages and Disadvantages

Though photogrammetry can provide data over a huge area, it is not the ideal surveying method in this project. The introduction of Digital Photogrammetric Systems brings improvement in

automated information extraction, but sensor orientation and image matching still need more manual editing to maintain the data quality (Baltsavias, 1999b). Baltsavias (1999a) summarized several advantages and disadvantages of using photogrammetric surveying developments.

Advantages of Photogrammetric Surveying

- The captured data can be downloaded, stored and kept in electronic format, such as tape, or disc.
- A huge amount of data is captured each surveying time.
- A small amount of labour input is good enough for the surveying.
- Three different dimensions of data are captured.
- Many well-developed commercial systems and custom-made products are available to use because of more than 30 years development history.

Disadvantages of Photogrammetric Surveying

- Hardware is really expensive.
- Detailed planning is required before the surveying, which takes time to plan.
- It is expensive to capture the data every time (including the labour cost, the flight).
- It cannot provide data fully automatically because of extensive manual editing for matching and reduction of the DSM to DTM even in digital photogrammetry.
- No photogrammetric surveying can be done during bad weather (raining or too cloudy) or at night-time. Therefore a clear and bright sky is required.

2.3 Light Detection And Ranging - LIDAR

Lasers (Light Amplification by Stimulated Emission of Radiation) were developed in the 1960's and have been used for geospatial sciences for many years. The laser is comprised of an emitting diode which produces a light source at a very specific frequency. The system sends a signal toward the target, and then it is reflected off a feature back toward the system. A receiver captures the return pulses. The distance between the laser system and the target can be measured by accurate timing. Laser based instruments emitted single laser pulses.

LIDAR (Light Detection and Ranging), the current laser technology, produces approximately 30-50,000 pulses per second. LIDAR has been used commercially since the early 1990s which is significant improvement over the last 15 years. Laserscanning emits many laser pulses at a time to scan across an area in a continuous motion. The laser pulses are emitted from the transmitter

and reflected from the object to the receiver . LIDAR is used to capture data on the terrain surface from an aircraft because of the advanced technolgy, GPS and inertial navigation system (INS).

Laser works similarly to GPS which calculates the distance between the transmitter and the target using the travel time of the pulses multiplied by the speed of the light. The distance (D) from the laser system to the object is equal to one-half of the light travelling time, v is the velocity or speed of light, and t is the time between emitting and receiving a particular signal, see equation 2.1.

$$D = vt \div 2 \tag{2.1}$$

There are two types of lasers used in LIDAR (Ackermann, 1999), pulse and continuous wave lasers. The pulse laser emits a narrow pulse in the near infrared region of the electromagnetic spectrum. Each discrete pulse is reflected from the Earth's surface and reflected to the receiver. The continuous wave laser emits a continuous signal stream and the receiver gets the full return wave. Phase measurements of the wave are used to calculate the distances.

The type of LIDAR system being used is based on the environment. A topographic system is used over land and works in the infrared portion of the electromagnetic spectrum. A bathymetric system is used over water to avoid signals being absorbed by the water.

The speed of light can vary in the atmosphere and the resulting distance can be affected by the variation in the speed of light. The variation can be modelled and corrected during the processing of the raw laser signal.

2.3.1 Airborne LIDAR System

Nowadays aircraft are used to take the measurements; therefore a LIDAR survey is called an airborne laserscanning survey (ALS). An ALS is a laser-based survey for the mapping of topography. It is a new independent and highly automated technology used to produce digital terrain models (DTM) and digital surface models (DSM) (Ackermann, 1999). ALS is being used more and more not only for research, but also commercially. ALS is commonly used to capture three dimensional data.

An airborne laser scanning survey is composed of three units: a laser scanning system, a global positioning system (GPS) and an inertial measuring unit (IMU). IMU navigates the altitude and the flight path of the aircraft which measures the 3 dimensional data. A high accuracy GPS is installed in the plane and a ground control base station is established. Fig. 2.4 shows an aircraft scanning a piece of land.

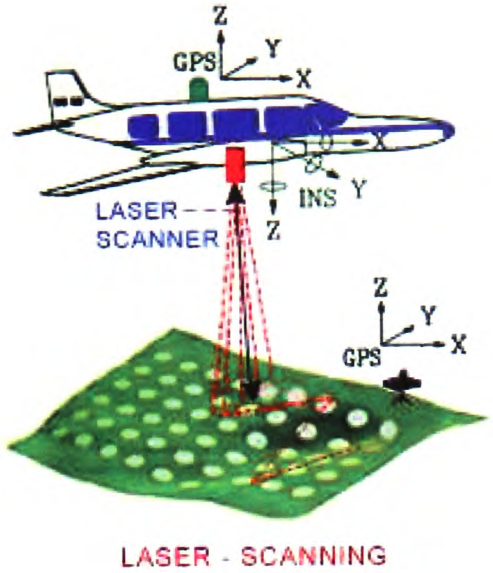


Figure 2.4: Airborne laser scanning

The laser scanning system is mounted on an aircraft or a helicopter just like an aerial camera. The scanner emits pulses toward the earth, and receives signals reflected from the earth's surface or other objects on the surface. The speeds of the measurements vary depending on the instrument but it can be up to 50,000 times per second.

A scanning mirror rotates transverse to the flight direction to collect the laser scan data. Normally the scan angle in both directions is less than 20 degrees from the nadir line, but some systems may scan up to 30 degrees. The laser scan signal forms a footprint on the surface, which is called an instantaneous field of view (IFOV). If the aircraft is completely level, the IFOV of the laser scan will be a circle in a vertical position. The IFOV becomes elongated to form an ellipse when the laser scan moves off the vertical. The IFOV enlarges along the scan direction and the distance between the scanner and as the ground surface increases.

However the signal also becomes distorted when it travels along the scan. The size of the signal on the ground will affect the density of a scan. The scan rate must be sufficiently fast to avoid any gaps between the data points. Different lasers emit signals at different rates, which are designated as kHz rates. For example, a 10 kHz system will generate 10,000 individual laser pulses in a second. A 50 kHz system emits 50,000 pulses in the same one second.

IMU and GPS are very important for determining the absolute position and orientation of the LIDAR sensors. GPS is a very popular technology which is used to search for the aircraft's

location. Ideally in a normal level flight, the laser instrument should capture the data vertical to the flight direction. However sometimes there will be inevitable variations and the inertial navigation system is used to correct the errors from the pitch, roll and yaw of the plane.

GPS consists of a series of satellites that emit signals in the radio frequencies and a receiver captures the signals to calculate the distance between the satellites and the receiver. More details about GPS can be found in Section 2.1. GPS is used to precisely determine the position of the scanner during the measurement. At least two receivers are used, one is on a known point on the terrain and the other one locates on the plane. The result is better if the on-board receiver is directly above the laser scanner. The relative position of the GPS receiver to the scanner will be accurately measured and the offsets can be used to calculate the scanner position.

The system has high accuracy gyroscopes which are used to measure the aircraft's attitude. The system measures the angular changes to allow the analyst to determine the orientation of the scanner. The IMU measurements are very important; therefore the angular change in all three directions must be accurately measured to make corrections.

Each of the above components is capturing data at set intervals; therefore timing is also an important issue. The location of the laser scanner needs to be interpolated between GPS and IMU sample times. Hence, the timing system is usually referred to as the fourth component of the ALS system.

There are two main reasons for post-processing of LIDAR data (Maune, 2001). The collected data needs a particular frame of reference. GPS is used to identify the location of the laser scanner and it is recommended that two ground receivers be set up over known points near the target area. The use of differential GPS can determine the receiver's location accurately based on vectors from two bases; therefore the captured data can be easily monitored for accuracy. The GPS system measures according to the World Geodetic System 1984; therefore the user has to transform the GPS coordinates to the required (local) coordinate system. With the position known from GPS measurement, the location of the return pulse can be determined by georeferencing.

The second reason for post-processing of LIDAR data is to eliminate the irrelevant data captured during the survey. Frequently users are interested in generating a bare-earth model. However, the laser scanner does not know whether the last returned pulse is the ground, the side or top of a building or somewhere else on the earth's surface. Different algorithms are required to separate the terrain from non-terrain objects (e.g. trees, buildings and bridges). Much of the filtering jobs can be done automatically; however still some man-made objects need to be removed manually and this is time-consuming.

A slant distance is captured when the laser scanner moves off the vertical position. The post-processing of raw LIDAR data after the initial aerial flight includes calculating the slant distance for each returned signal. This data is corrected for atmospheric effects. The IMU system captures

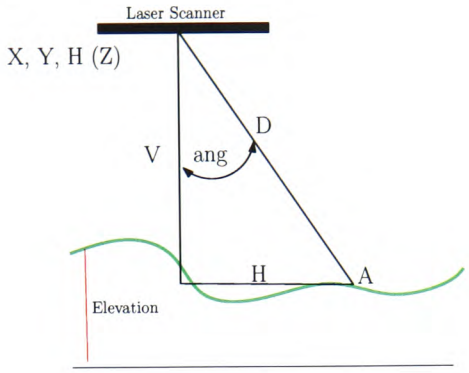


Figure 2.5: Geometry of the laser scans pulse (Burtch, 2002)

the angles of the flight direction when roll, pitch and yaw occur during data collection, and the corrections are applied to the sensor orientation. The GPS data are processed separately and then imported into the LIDAR processing system. The elevation of the ground point is calculated using the position of the sensor and the swath angle in each individual scan (Fig. 2.5).

Let's assume the laser signal was sent out at a 10 degree angle between the vertical (V) and the slant distance (D) and assume that the sensor orientation is perfect with no roll, yaw or pitch effects. The slant distance between the scanner and the ground surface is 1,100m. Using simple trigonometry, the vertical distance between the sensor and the ground when recording the returned signal from position A is:

$$V = D * cos(ang)$$
$$= 1,100m * cos10^{o}$$
$$= 1,083.29m$$

If the GPS on the aircraft measured the location of the sensor at the time when the signal was sent, the local coordinates and the vertical height of the plane are:

$$X_{sensor} = 1,800,800.30m$$
$$Y_{sensor} = 600,600.50m$$
$$Z_{sensor} = 1,400.20m$$

The ground elevation of point A is

$$Elev_A = H_{sensor} - V$$

$$= 1,400.20m - 1,083.29m$$

$$= 316.91m$$

For example, the horizontal distance, H_A, from the vertical line to the ground point can be computed using basic trigonometry.

$$H = D * sin(ang)$$

which for the distance between the vertical point from the sensor and location A is:

$$H = 1,100m * sin10^o$$

$$= 191.01m$$

The aircraft is assumed to fly from due north (along the Y-axis) and the scan angle is to the right (to the east of the vertical line), then the Y-coordinate would remain the same and the X-coordinate would become

$$X_A = X_{sensor} + H$$

$$= 1,800,800.30m + 191.01m$$

$$= 1,800,991.31m$$

The x and y coordinates of the ground point A are 1,800,991.31m, 600,600.50m and the elevation of point A is 316.91m. Each return has been georeferenced; therefore it is easy to determine the location of the captured data points. Though the theory of georeferencing seems simple, it is somewhat more complex in reality. Mathematically, the calculations are handled easily within the processing system.

When sending a signal to the Earth it can hit more than one object, the first signal may hit the foliage of a tree and the rest hits the ground. The sensor can be set to collect all these data pulses simultaneously. They are commonly called the first pulse (the first returned signals are normally from trees or vegetation) and the last pulse (the last returned signal hitting on the ground or the buildings). Some systems are able to collect up to 5 different returns.

Though some of the signal of LIDAR is able to penetrate tress and hit the ground surface, it is not possible to see through trees, especially for the fully closed canopies (e.g. rain forests).

2.3.2 Accuracy of the ALS System

Currently, a number of commercial LIDAR systems make optimistic claims as to the accuracy of LIDAR data which is 15cm vertically and 0.3-1m horizontally (Flood, 2001; Fowler, 2001). The accuracy of LIDAR data depends mainly on the three components of the system (laser scanner, GPS and IMU) which are affected by; the accuracy of the various instruments, GPS, the flying patterns, flight layouts, flying time and ground control. The estimated accuracy assumes that the system is properly calibrated and functioning correctly and that the condition of the earth's surface is ideal.

Some rules that affect the accuracy of the LIDAR (Brinkman and O'Neill, 2000) include:

- The slower aircraft captures denser spot spacing.
- More reliable or accurate DTM is produced using denser spot spacing more data points are collected.
- Vertical accuracy decreases when the scan angle increases.
- The intervals between the laser pulses affects the density of the data points.

2.3.3 Advantages and Disadvantages

Several advantages and disadvantages of LIDAR are discussed in this section. Some of the advantages make LIDAR popular. Many studies have found a lot of use for LIDAR in terrain mapping. Though it is costly, it offers high precision and high point density data for DTM modelling.

Advantages of LIDAR

LIDAR is very versatile. It has been used for atmospheric studies, bathymetric surveys, glacial ice investigations, and many other applications. Each survey measures a huge amount of data which is more efficient than using GPS or traditional surveying instruments to take measurements point by point. Besides efficiency, it has the ability to map topography in remote, difficult to access, or environmentally sensitive areas, for example, salt marshes, forests and woodland, and cliff coastlines.

LIDAR is also considered as a superior data collection tool over photogrammetric survey because it is really difficulty to identify terrain points (easy enough to recognise) in the imagery (e.g. forest or desert).

The high precision and point density of LIDAR data provides a really good data source for DTM modelling. Brinkman and O'Neill (2000) thinks that LIDAR data can accelerate the project schedule up to 30% because it can be used for DTM modelling almost immediately.

LIDAR survey is direct terrain data acquisition. The measured results of LIDAR gives the 3D positions of terrain points which are convenient for further development. Unlike photogrammetric surveying it involves a number of post-processing steps to extract 2 and 3D information from the surveyed photos.

One of the advantages of LIDAR over photogrammetric surveying is that it can be done during day or night. Theoretically LIDAR is an active system which can be used 24 hours a day, however it cannot be used above cloud cover or when fog, smoke, mist, rain or snow storms are present. The turbulence of the plane will cause problems with the inertial system; therefore it is better to avoid surveying during high winds.

LIDAR requires that only one reading penetrates the trees to reflect from the ground; however photogrammetry requires that the same ground point be visible from at least two aerial photos. This cuts down the area which can be identified for producing a contour map.

Disadvantages of LIDAR

However LIDAR is not a cheap data source. The cost of doing a laserscanning survey includes not only the laserscanning instruments, as well as flying an aircraft with an expert. Small companies cannot afford to do laserscanning by themselves. They may need to purchase data from big companies. Since the technology can improve quickly, the cost of the data collection may decrease in the next two to three years.

An advantage of LIDAR is able to do bathymetric surveying; however the disadvantage is not easy to separate water boundaries from land using LIDAR by itself. Laser scan data are normally collected in a regular spacing pattern which is not possible to target on a specific feature. For example with spacing of 5 meters scanning space, it is not possible to survey a 2 meters feature. No standards have been established to monitor the quality of the results, but a number of studies are being undertaken to alleviate this problem (Burtch, 2002).

The data includes all information on the terrain because the laser signal hits all the objects on the ground and returns to the machine. It is not an easy task to identify or filter all the surveyed objects. To identify buildings or to generate a bare-earth model are interesting topics. One of the tasks in this research is to recognize building blocks from raw LIDAR data.

2.4 Conclusions on Reviews of Data Sources

LIDAR is an ideal data source for this research because it provides an efficient capture of high resolution digital surface models without manual editing. Though a GPS surveying has high accuracy, it takes a long time to survey a piece of land. A photogrammetric surveying captures a large area but LIDAR surveying seems to have fewer limitations, for example post-processing and the weather.

As mentioned in Section2.3, there are many companies who provide LIDAR data. Many research developments, based on LIDAR data, include 3D mapping, map updating, forestry and so on. However LIDAR systems produces only x, y and z coordinates and no additional information (Ackermann, 1999). As I am interested in using LIDAR for 3D city modelling, the next chapter will show some existing research methods for extracting building information from LIDAR data.

2.5 Chapter Summary

Three surveying methods are described which include GPS, Photogrammetric and ALS surveying. Each has its advantages and disadvantages. This project requires a surveying method which captures a large amount of data at a time which is easy to obtain and maintains the originality of the captured data. After comparing three surveying techniques, LIDAR is the most suitable data source for this project.

GPS surveying is taken by putting a receiver in a specific position to capture and calculate the signals from satellites. Of the three methods, GPS gives the best accuracy; however data points are captured individually which is time consuming and labour intensive. When a surveyor needs to carry out a survey in a private area, he needs permission to do it which can be an obstacle for the surveying. According to the properties of GPS surveying, it is not ideal for this research.

Photogrammetric surveying mounts a digital or traditional camera on a plane and flies across the terrain surface to take photos. Each flight captures a large number of aerial photos with digital cameras. The photos are analysed to extract 3D data points which takes extra processing time and decreases accuracy of the data. Due to these disadvantages using photogrammetric data is not suitable for this project.

ALS mounts a laser scanner on a plane and scans across the terrain surface to capture 3D data points by emitting and receiving the returned laser pulses from the terrain objects. The laser scanner emits two pulses at a time to the surface, the first pulse reflects signals from the surface of objects (e.g. the tops of trees), and the second one penetrates the vegetation and returns signals from the ground surface. In each survey a huge number of data points are captured

in a three dimensional data format; therefore no further post-processing is needed to extract 3D information. The LIDAR system emits and receives light pulses reflected from an object; therefore the survey can be done in day or night. LIDAR is the most suitable data capturing method for this project.

LIDAR captures three dimensional data points; however no additional information comes with the captured data to describe the terrain objects, for example location of the vegetations, building boundaries, and roof structures. The data looks like a bunch of point cloud; therefore further processing is needed to extract useful information. This project is interested in using the last pulse LIDAR data points to reconstruct three dimensional buildings. The next chapter shows some current research on extracting and reconstructing buildings from LIDAR data.

Chapter 3

General Background - What is the Problem?

The last chapter shows the reason of choosing LIDAR for this project. This chapter reviews some current building reconstruction methods which includes filtering, building outline extraction, roof modelling and building reconstruction. The reasons of looking for an alternative approach shows in the chapter summary.

The up-to-date airborne laser scanning surveying always produce a huge number of data points. The point cloud is composed of signals returning from the Earth's surface which includes man-made and natural ground objects (e.g. buildings and trees). However, feature (e.g. building and forest) extractions from the point cloud is a labour-intensive task. A manual classification with quality control produces a DEM and 60-80% of processing time is the human effort (Flood, 2001), which is really time-consuming.

Many post-processing algorithms have been developed to combine additional data sources such as intensity imaging (Forlani et al., 2003), multi-spectral imagery, or GIS data (e.g. cadastral data or ground plans) (Haala and Brenner, 1999). However, Forlani et al. (2003) admitted that the extraction of maximum useful information from a single data source still plays an important role.

Display technology, driven by the games development business, is now satisfactory even on modest computers. The data structures connecting the various portions of the buildings and terrain have been well defined in Computer Aided Design (CAD), Computational Geometry and Geographic Information Systems, although they have only recently been integrated. The greatest remaining difficulties are efficient data collection and modelling. Though many available surveying methods provide efficient and high accuracy data, not all of them can provide rapid elevation models of terrain and buildings. The remaining challenge is to convert the captured data into CAD-type models containing walls, roof planes and terrain which can be rapidly displayed from any 3D viewpoint. The best method will be an automated procedure without any manual operation.

3.1 Algorithms for Generating a Bare-Earth Model

In airborne laser scanning data, filtering is normally used to remove man-made objects to obtain a bare-earth model. Order statistics and morphological filters or weighting functions are often used to distinguish points on the buildings or tree canopies from those on the ground. Buildings are reconstructed on a bare-earth model; therefore filtering is an important process. Over the past years, many filter algorithms have been developed. Sithole and Vosselman (2003) published a comprehensive report comparing filter methods. They concluded that filtering algorithms generally have a better performance on specific surfaces and the performance may be affected by the complexity of the ground objects and the point density. To improve the quality of filtering, they suggested the use of additional information to select suitable filter algorithms depending on the type of landscape.

The following paragraphs show twelve filtering algorithms which include:

- Segmented filter algorithm; Jacobsen and Lohmann (2003) divided the laser scanning points into different groups and then filtered the data in each group individually to optimise the filtering performance. They segmented the data into eight general classes based on the height combined with the intensity. The results will be improved when the classes are filtered individually.
- Elmqvist (2001) and Elmqvist et al. (2001) developed an algorithm to estimate the ground surface using active shape models, which matches a deformable model to an image by means of energy minimisation. When applied to LIDAR data, the active shape model acts like a membrane floating up from underneath the data points. An energy function is determined by analysing the way in which the membrane will stick to the data points. The energy function is the smallest when the membrane sticks to the ground. The active shape model is a weighted combination of internal and external forces, which come from the shape of the contour and the image (laser data) respectively. This is a robust algorithm, and Elmqvist used a plane below the lowest point in the data set as a starting place.
- Sohn and Dowman (2002) developed a regularised method first to divide the entire terrain surface into a set of piecewise segments so that they have "homogeneous" background knowledge of the underlying terrain slope as being "plane". Then a criterion is set to label the on- and off-terrain points from the terrain. The next step is to seek the "best fit" planar surface model to reconstruct underlying pieces of terrain planes by checking intra- and inter-relationships between the labelled on- and off- terrain points. An initial terrain surface model is prepared that covers all the LIDAR points. The downward divide-and-conquer methods reconstructed a set of the planar terrain models locally. Then the upward divide-and-conquer triangulation is triggered over a local area, a set of the tetradhedron model candidates is generated and the minimum descriptioni lenght criterion (Rissanen,

1984) is used to chose the best one. The process ends when all the terrain models are checked.

- Morphological filtering algorithm is developed by Vosselman and Maas (2001) which is based on height differences in a representative training dataset. Filter functions are derived that either preserve important terrain characteristics or minimise the number of classification errors.
- Roggero (2002) modified the morphological filter developed by Vosselman (2000) and Vosselman and Maas (2001). The algorithm uses the linear regression criterion to estimate the bare-Earth. A curve is obtained from these maximum heights above the regressed line which presents the initial bare-Earth. Points are classified as bare-Earth, object or unclassified after the bare-Earth has been determined.
- Brovelli et al. (2002) believed objects in a landscape stand out from the bare-Earth and they have edges to form a closed boundary. The point cloud is tiled and 200 times 200 splines are established in each tile. The splines are interpolated on the data. Points below and above the splines are potential objects or bare-Earth; therefore edges (boundaries) are found between the object and bare-Earth points. Points are accepted as object points when they are inside the boundary and their height is equal to or greater than the mean edge height. The algorithm needs to input the first pulse data and gives a grid system as an output.
- A Hierarchical Modified Block-Minimum Method was developed by Wack and Wimmer (2002). This used a hierarchical approach and block-minimum algorithm to detect non-terrain raster elements. The algorithm generates a 9m raster DEM from raw data points which is used to overcome large buildings and dense vegetation. The lowest height value is used to compute all the points. Because of the size of the DEM, most buildings and dense vegetation should be filtered out. Repeated procedures with the use of a Laplacian of Gaussian (LoG) operation to remove all the share elevation changes in the landscape. The result of DEM is used to generate a 3m DEM. At a resolution of 3m and below DEM, a weight function that considers the standard deviation of the data points and the LoG operation are used to generate a Bare Earth. The result DEM is used to obtain a lower resolution DEM, for example, 1m DEM and so on. User intervention is needed to obtain a good result when setting the best parameter in the determination of the initial 9m DEM.
- Pfeifer and Stadler (2001) used a hierarchical method and robust interpolation (in each hierarchy level) to classify the original points and to derive the terrain. An initial roughly approximated terrain is computed using the points of respective hierarchy levels. A weight function is used to assign weights (vertical distance) to points above (small weight) and below (large weight) the approximate surface. The surface is recomputed using a linear interpolation function and the assigned weights. The process is iterated until the surface has an insignificant change between iterations or certain number of iterations have been reached. To complete the iteration, points vertically above and below the surface within a

predefined threshould are accepted as bare-Earth. An extended robust interpoltation, the hierarchic robust interpolation (Pfeifer and Stadler, 2001) to refine the accepted bare-Earth model.

- Bretar et al. (2004) classified the LIDAR data using a multiple pass filter. The algorithm is based on a bipartite voting process. Each laser point was labelled either as a ground or non-ground point several times and the point finally classified by the most frequent label. Then a propagated mechanism considered the moving direction and a ground surface was estimated. Finally an energy minimization algorithm was used to refine an initial DTM.
- Axelsson (1999, 2000) used a Minimum Description Length criterion to classify buildings and electrical power lines from the ground surface. A sparse TIN is derived from neighbourhood minima, and the laser points are added to the TIN progressively. The algorithm creates a TIN and points are added to the TIN model if they are below data derived thresholds. The threshold is recomputed at each time of the iteration. The iterative process ends if no more points are below the threshold.
- Vosselman (2000) developed a slope-based filter which classifies ground points by the gradient between the natural slope of the terrain and the slopes of non-terrain objects. The filter was modified so that the threshold changes with respect to the slope of the terrain. Sithole (2001) modified the slope-based filter to overcome the limitation of gentle slopes. Sithole (2001) and Oude Elberink and Maas (2000) suggested to separate urban and vegetated areas during filtering and different parameters were used in urban and vegetated areas.
- Oude Elberink and Maas (2000) developed a method which aims at segmenting raw laser scanner data in an unsupervised classification using anisotropic height texture measures. The operations can separate oriented and non-oriented objects: however it needs high-density laser scanner data. It returns poor classification results for roads, grasslands and agricultural fields when the reflectance measurements are poor quality.

3.1.1 Problems with Current Filtering Algorithms

Many current filtering algorithms aim at removing non-terrain objects (e.g. trees and buildings) and generating a bare-earth model, so that the bare-earth model can be used with visualising tools for the landscape or as a base for further modelling (for example city modelling with roads and buildings). Most of the algorithms do not give a perfect result because it is not an easy task to remove objects in different environments. Some algorithms work better on forests and some on urban areas Sithole (2001) and Oude Elberink and Maas (2000).

The most important reason for not using filter algorithms is to avoid losing any important information during the filtering process. A bare-earth model is not required to reconstruct

buildings because buildings are extruded from the terrain surface instead of being pasted on the surface in this project. Details will be shown in Chapter 6.

3.2 Algorithms for Building Extraction

Airborne laser scanning captures natural and man-made objects on a surface. Natural objects include trees, rivers, lakes and so on. Buildings are one of the most important artificial objects on the ground. Additional data sources have been used to extract buildings in many algorithms. Building extraction determines building locations, height, size, orientation, roof style, etc.

Building extraction can be either automatic (without any human input) or semi-automatic (with certain amount of human input). Normally building extraction includes looking for roof structures. Data-driven and model-driven approaches are commonly used. The data-driven approach extracts the building shape using the data without any pre-defined building models. The model-driven approach finds the best pre-defined building model to fit in a building footprint. Some algorithms analyse the raw LIDAR data directly or process grid-based images from LIDAR data, and some use both data structures at different processing stages.

Several building extraction and modelling methods are described in the following. Most of them combine additional data sources (e.g. aerial images, 2D ground plans) to extract and remodel the buildings.

3.2.1 Building Extraction from aerial images

- Brunn and Weidner (1997b) used a model-driven approach to reconstruct building models. Binary classification and Bayesian Networks for classification were used to separate buildings and vegetation from a digital surface model. Roof extraction from DSM aimed at the extraction of polyhedral building models which included two subtasks: roof plane detection and roof plane reconstruction. Parameter estimation and the derivation of hypotheses were used in roof segmentation and reconstruction. The limitation of the roof extraction was due to the use of polyhedral models. In 2001 Brunn (2001) showed that using statistical interpretation to extract buildings would be an interesting field for future work. The research included statistical detection of buildings in multi-sensorial data, the reconstruction of buildings from a single image, and the refined reconstruction and interpretation of buildings using localised building models.
- Bonn (1999) used an automatic approach to extract buildings from aerial images. Oxford's approach (Baillard et al., 1999; Baillard and Zisserman, 1999) and Bonn's approaches (Fischer et al., 1998) were used to extract buildings with known orientation. Oxford's approach formed 3D edges from 2D line segments using a line matching algorithm based on

the texture of the roof to create the half planes model, and finally by using the polyhedral faces triggered the matching algorithm to paste the building image onto the model. Bonn's approach was based on the CSG-description of buildings. First building corners were extracted, and then building parts were described by hypotheses. Finally parts of buildings were aggregated between pairs of basic buildings.

- Gerke et al. (2001) used aerial colour infrared images and a digital surface model as the source of information to extract building information. He made use of colour infrared images and height models to locate possible building areas. Then the information about the surroundings of a building was used to locate individual buildings. Finally an orthogonal closed polygon presenting the roof outline was rotated and verified using invariant geometric moments and then reconstructed as an orthogonal geometric model.
- Multi-source building extraction was presented by Mikhail (2000). DEM and thematic data were used to support the building extraction. The extraction used a combination of tools, reconstruction and reasoning in 3D, use of multiple sources of data and perceptual grouping. The building detection system was based on the "hypothesis and verify" paradigm, which functions with a pair or more of panchromatic (PAN) images. Fragmented lower level image features made the hypotheses for potential roofs. Then promising hypotheses were selected and verified by global evidence (e.g. walls and shadows). Further verification of the selected hypotheses was required to form a building. The Bayesian reasoning approach was used to help decide whether the result of the evidence was accurate enough to put the building in. Mikhail suggested using additional data sources to improve the results, for example DEMs, LIDAR or multi- or hyper-spectral images.
- Suveg and Vosselman (2004) used maps (ground plans) and combined aerial images for building reconstruction. The process was described as a search tree which was generated by multiple, consistent, building primitive hypotheses in a CSG model representation. First the ground plan of the building was used to partition the building into simple parts. A building can usually have multiple partitioning schemes. Minimum Description Length (MDL) was used to rank the partitioning schemes in the second level of partitioning. The next step is to check all the possible height values of each partition. The final level generated building hypotheses for each building partition. The building hypotheses were refined, verified and matched to the image itself or the information was extracted from the image. The system was able to reconstruct more than 75% of the buildings in the test area.
- Henricsson and Baltsavias (1997) used aerial images to extract buildings. The approach started with a given region. 2D line segments were extracted, and then the 3D information was computed using stereo matching of the segments. Coplanar grouping and similarity grouping methods were used to find the planes for building reconstruction. The approach made use of the colour information during stereo building extraction.
- Brenner and Haala (1998); Haala and Brenner (1999) used a model-driven approach and

additional 2D ground plans as data sources for an automatic or semi-automatic building reconstruction approach. First aerial images were used to classify between building and vegetation areas. They believed buildings were represented by a combination of one or more basic primitives; therefore 2D ground plans were then used to divide the buildings into rectangular primitives. Each primitive came with a defined position, orientation and horizontal extension. Unknown parameters like the height, roof types and roof plane slopes were estimated by a least squares adjustment. A number of pre-defined 3D parametric primitives were fit to the DSM surface.

3.2.2 Building Extraction from Airborne Laser Scanning (ALS) data

- High resolution of LIDAR data were used to extract and reconstruct buildings by Forlani et al. (2003). The system was based on grided LIDAR data. Initial segmentation separated the terrain and non-terrain features. A hierarchical set of aggregation rules were used to locate building areas. Certain potential buildings were further segmented as roof planes based on gradient orientation and plane fitting by a commerical algorithm (RANSAC) (Fischler and Bolles, 1981). Finally buildings were formed by joining along the eaves lines and the roof edges. A CAD system (Autodesk) was used to visualise the reconstructed buildings.
- Morgan and Tempeli (2000) and Morgan and Habib (2002) presented automatic building extraction methods. The approach started with re-sampling the laser data onto a regular grid. The first step of their approach was to cluster the laser points with their common characteristics. A morphological filter was used to classify the laser points which were based on the height values within a search window. Then the terrain and non-terrain segments were identified, the non-terrain segments were connected to form larger segments (building hypothesis). The algorithm to extract the building geometrical parameters had two main steps, which were: determining the 3D internal building breaklines and estimating the building boundaries. A region-growing method based on least-squares adjustment was used to extract building facades from laser data connected by a TIN model. The final result was a 3D vector representation of the buildings.
- Wang and Schenk (2000) used an edge-based method to detect the building outline from LIDAR and a TIN data structure to reconstruct the buildings. The whole process took terrain surface data as input and proceeded with edge detection, edge classification, building point extraction, TIN model generation and building reconstruction. Building outlines were classified and distinguished from other edges, which were based on geometry and shapes with orthogonality, parallelism, circularity and symmetry. Building points were clustered and each group of building points was triangulated into piece-wise planes. Tri-intersection of every three planes derived the building corners and their relative orientation. The study area included nine buildings from large to small sizes, and contained different

roof patterns (e.g. peaked roof, flat roof and multiple level flat roofs).

- Tarsha-Kurdi et al. (2007) chose RANSAC and extended to exceed its limitations to detect the best mathematical plane among 3D building point cloud to search for roof planes. RANSAC algorithm is used to detect mathematical features like straight lines and circles which many authors used the algorithm to detect roof planes (Brenner, 2000; Forlani et al., 2006). The paper presented that RANSAC algorithm detects roof planes rapidly and accurately.
- Elaksher and Bethel (2002a,b) used the geometric properties of urban buildings to reconstruct from LIDAR data which was a grid-based method. The algorithm started by looking for potential building points that are used to populate a plane parameter space. A minimum filter is used for the process of finding the candidate building points. When the plane parameter space is filled, the next step is to find the planes which can represent the building roof surfaces. DEM points were clustered to different planes. A region-growing algorithm was used to complete the roof extraction. Further refinement of the extracted roof region included applying geometric constraints to building roofs. The roof regions are used to find the roof region border points. The algorithm from Bimal and Kumar (1991, 1992) is used to convert the roof border points to polygons. Finally the building wire-frames are formed by the building corner points.
- Hofmann et al. (2003) and Hofmann (2004) developed a data-driven method for automatic building reconstruction from raw ALS data which made use of the properties of a TIN-structure. Data points were mapped onto a triangular-mesh, which were then clustered using an agglomerative approach. After analysing the clusters, significant roof planes were extracted. Roof structures were formed by intersecting and trimming the roof planes. With the use of the ground plan, the building models were created following the top lines of the walls. Similar research was done by Gorte (2002) who used a TIN-structure to segment the laser scanner data. He compared the TIN segmentation with 3D Hough transformations (Maas and Vosselman, 1999) which were simpler, had fewer parameter to control, ensured connected planes and were capable of handling a huge amount of data.
- Maas and Vosselman (1999) and Maas (1999) developed two techniques for extracting building models from raw laser altimetry data. Based on invariant moments, a closed solution for the determination of the parameters of a standard gable roof type building model was achieved when a rectangular ground plan was found. Asymmetric deviations like dorms on roofs can be modelled. A data driven approach to detecting planar roof faces used triangulated point clouds, and intersect the faces to form more complex buildings. The roof or the building outlines were determined by extracting the connected components of triangles that belong to the roof surface from the triangulated and segmented laser points. The extracted buildings looked regular but many model constraints that were part of the invariant moment approach are not fulfilled.
- Very high density height data were used to reconstruct buildings (Vosselman, 1999a).

There are two assumptions in the process. The first assumption is that buildings are described by a number of planar faces. The 2D Delaunay triangulation was used to determine the planar faces of the roofs. When the size of the largest connected component of the selected triangles is greater than some threshold, then a planar face has been found. The contour of the connected component is extracted for reconstruction. The second assumption is that all the edges with height discontinuities are either parallel or perpendicular to the existing main building orientation. To determine the building outline, the first step is to intersect the adjacent planar faces to find the ridges and valleys. Then determine the building outline where the second assumption is made. The third step is to find the roof surface discontinuities. Finally building walls are created below all lines with height discontinuities. The lowest point around the building determines the lower line of the walls.

- Vosselman and Dijkman (2001) combined point clouds and ground plans to reconstruct 3D building models. They used a 3D version of the well-known Hough transform to extract roof planes. They recommended that all detected planar faces should model some part of the building which made it possible to reconstruct more buildings and more details of the buildings.
- A computer vision based approach was introduced (Babu Madhavan et al., 2006) to model buildings from laser scanning digital surface model. The system started with a pre-processing procedure to extract a DSM for spatial resolution enhancement. It was used for stable planar regions and edge extraction. Stable planar region extraction included computing the histogram of the local surface normal vector, hierarchical plane extraction and region merging. The jump and boundary edges were extracted by applying Sobel directional filtering and a sequential Hough transform. Roof edges were found using the geometries of the planes obtained because any two adjacent regions determine the roof edge. Finally a building was described as a polygon with computed roof edges and the line segments of the jump and boundary edges which can be turned to a 3D building.
- Hu (2003) suggested using ALS data to extract buildings. His approach started by generating a digital non-terrain model (DNM) and vegetation support model (VSM). First the DNM is classified to locate the object footprints and then potential buildings were found by classifying their shapes. The building boundaries were found using an enhanced Hough transform and a sequential linking technique. He tried to classify the ridge vertex for roof modelling. The three types of vertices were: I-vertex (visible with only one boundary vertex), V-vertex (visible with two adjacent boundary vertices), and Y-vertex (visible with three consecutive boundary vertices). His research mainly detected the building boundaries, and the roof modelling method was not really successful.
- Rottensteiner and Briese (2003) used high-resolution LIDAR data only for building extraction and the binary image was used to detect the planar patches. Though the process had been improved from semi-automatic to automatic, pre-defined building models were

still necessary for building reconstruction. The data structure of the primitives and the building models was based on B-rep (boundary representation) methods in the system.

- Shen (2008) used an Alpha shapes algorithm to extract building boundary from LIDAR point clouds data. Improved Pipe algorithm and boundary regularisation are used to refine the building boundary. His algorithm shows an accurate method to extract building outline; however the roof structure is not considered in the method.
- Zhang et al. (2006) extracted building footprints automatically from airborne LIDAR measurements. The algorithm starts with separating ground and non-ground objects using a progressive morphological filter. A region-growing algorithm and plane-fitting technique are used to identify building objects. Then a raw building footprint can be obtained by connecting the boundary points. Finally the raw footprint is further simplified and modified to remove noise using the Douglas-Peuker algorithm. He presented a highly automated method to identify the building footprint, further development is required for remodelling the roof sturcture.

3.2.3 Building Extraction from Aerial Images and ALS (LIDAR) data

- Gamba (2000) developed an algorithm to extract 3D buildings from DSMs and implements the algorithm using LIDAR and interferometric radar (IFSAR) data. He aimed at creating an automated algorithm which extracted information on urban structures from medium or high resolution data. The method was based on Jiang and Bunke (1994)'s scan line segmentation method to extract planar surfaces from the grid data. It grouped the pixels into different segments which were used to find the seeds. The seeds and segments were used to find the planes these were used to characterize the original image. If the planes had similar characteristics or were adjacent to each other: they could be grouped. Because of noise not all the pixels belong to a plane. The algorithm seems to work better on extracting relatively large, isolated structures. He suggested combining two data sources for the building extraction which may give a better result.
- Sohn and Dowman (2001, 2003, 2004) believe that combining two different data sources for building extraction provide a better result. They used high resolution satellite data (e.g. IKONOS images) combined with LIDAR to extract buildings. A building was assumed to be a generic polyhedral shape with some degree of geometric regularity. They believed a building was aligned piecewise with its neighbouring building; therefore the Building Unit Shapes (BUS) were extracted to form the building boundaries. First, straight lines were extracted from IKONOS images, and then filtered and regularised to form a preliminary building boundary (BUS). When the initial building outline was found Binary Space Partitioning and heuristic filters were used to modify and finalize the building boundary. A statistical homogeneous predicate test is used to extract building boundaries and the segmentation method can be used to determine the roof tops in his further research.

- Rottensteiner (2001); Rottensteiner and Briese (2002) combined LIDAR and aerial images to extract building footprints. LIDAR DSM and DTM were used to detect building footprints. Then aerial images were used to generate initial planar segments. A set of simple primitives was used to reconstruct buildings using Boolean operators. In 2007, Rottensteiner et al. (2007) introduced the Dempster-Shafer fusion to detect buildings. It gave a better result when ALS data and multi-spectral images were used for the detection. The algorithm correctly detected 95% of buildings with areas larger than $70m^2$ and improved the accuracy of small residential buildings by up to 20%.
- Khoshelham (2005) integrated aerial images and LIDAR height data to reconstruct and refine buildings and roofs. An image was partitioned into a number of homogeneous regions which corresponded to surfaces in space. Then the regions were refined based on fitting planar surfaces to the height points that project into each region. The region refinement included plane fitting and split-and-merge processes. The refined regions were used for roof reconstruction. The advantages of this method included a higher level of robustness during reconstruction and the reconstructed parametric roof planes were more effective for the final modelling step. He suggested integrating the two data sources to improve the region partitioning and refinement results.

3.2.4 Problems on Current Building Extraction Algorithms

Building extraction seems to include building detection and reconstruction; however the two processes may not be clearly distinguishable (Hu, 2003). Some approaches use interpolated LIDAR data (e.g. grid or DEM) to extract buildings. Loss of data or important information can be avoided if original raw LIDAR data is used.

Combining airborne laser scanner data and aerial images for building detection seems to have promising results (Rottensteiner, 2001), for example some approaches make use of colour information from the images (Gerke et al., 2001; Henricsson and Baltsavias, 1997). An additional data source such as ground plans or aerial images may improve the quality or efficiency of the extraction; however LIDAR data is the only data available for this research.

A model-drive approach seems really popular (Brenner and Haala, 1998; Haala and Brenner, 1999; Brunn and Weidner, 1997b; Rottensteiner and Briese, 2003; Rottensteiner, 2001; Rottensteiner and Briese, 2002). However it takes time to create a library of building models, and is less flexibile on the building shapes.

The approach of this research is based on a TIN-structure and then the topological connectivity is preserved during the whole process. The approach starts with creating a TIN model using the raw LIDAR data and analysing the properties of the triangles in the model. This looks similar to Vosselman (1999a); Forlani et al. (2003); Wang and Schenk (2000); Elaksher and

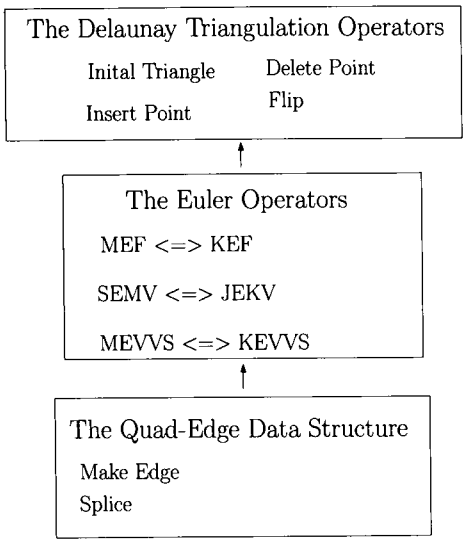


Figure 3.1: Three level of Operators

Bethel (2002a,b); Hofmann et al. (2003); Hofmann (2004); Gorte (2002); Maas and Vosselman (1999); Maas (1999). However region growing method is not used to search for the roof planes which avoids any discontinuity of the triangles. The details are in Chapters 4 and 5. Buildings are extruded from the terrain surface instead of pasting them on top of the bare-earth model.

3.3 Algorithms for Building Reconstruction in this Project

One of the main objective of the project is to keep the topological connectivity while reconstructing 3D buildings from LIDAR data. Tse and Gold (2003) proved the use of the Quad-Edge data structure (as the basic data structure), the Delaunay Triangulation, and the Euler Operators (as the reconstruction and operational tools) to create 3D buildings which preserved the topological connectivity.

Tse (2003) described the reason for using the Quad-Edge data structure, the Delaunay Triangulation, and the Euler Operators in this project. We have been successful using the CAD-type Euler Operators to extend a TIN model (Tse and Gold, 2001, 2002a, 2004, 2003). The idea is based on three levels of operators which are; the lowest level (the Quad-Edge data structure), the middle-level (the Euler Operators), and the highest level (the Delaunay triangulation) in Fig. 3.1. The details of the three-level operators are shown in the coming sections.

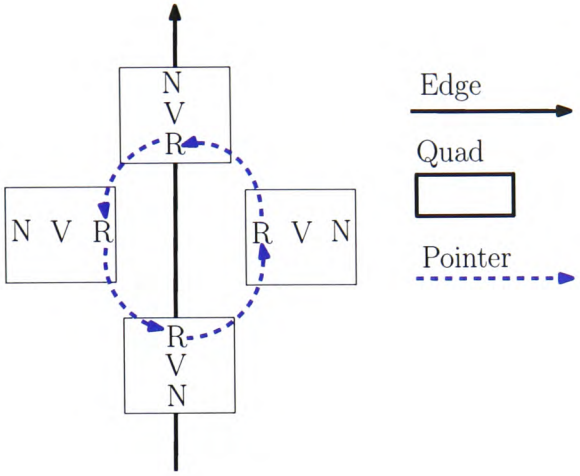


Figure 3.2: Quads in an Edge

3.3.1 The Quad-Edge Data Structure

The Quad-Edge data structure (Guibas and Stolfi, 1985), which is the lowest level operator, consists of three elements: vertices, edges and faces. Edges take the leading role because they store the complete topological information about the polygons. The following details each element:

1. Edges represent directed edges, and each edge has pointers to the next and previous edges.
2. Vertices store geometric information with the coordinates (e.g. X, Y, Z).
3. Faces are triangular in shape and formed by three edges. The edges in each triangle connect in anti-clockwise direction.

Each edge has four "Quads". Each Quad has three entities; Vertex (V), Next (N), and Rot(R) in Fig. 3.2. V stores the coordinates of the edge vertices, and the centre points of the left and right triangles. N connects to the next edge in an anti-clockwise order. R connects to the next 1/4 of the edge (one Quad) in anti-clockwise direction. Points "pt1" and "pt2" are the origin and destination (start and end points) of Edge "e" in Fig. 3.3 respectively.

Two simple topological operators in the Quad-Edge data structure are "Make-Edge" and "Splice" in Fig. 3.4. Make-Edge makes an individual unconnected edge in Fig. 3.4(a) which create four connect quads in Fig. 3.4(a). Splice connects or disconnects the edges. It also splits or merges the "Next" loops. It merges two vertices by splitting a face into two pieces (from left to right of Fig. 3.4(b), or it splits a vertex by merging two faces (from right to left of Fig. 3.4(b)).

Edges can be topologically or geometrically connected. They can have the same origin geometrically but not be topologically connected. Splice topologically connects or disconnects the edges. Splice for two unconnected edges connects them and splits the face between them. Splice for

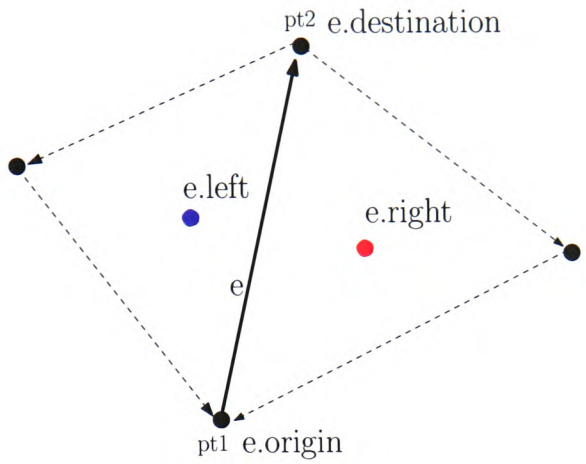


Figure 3.3: Navigation Operators

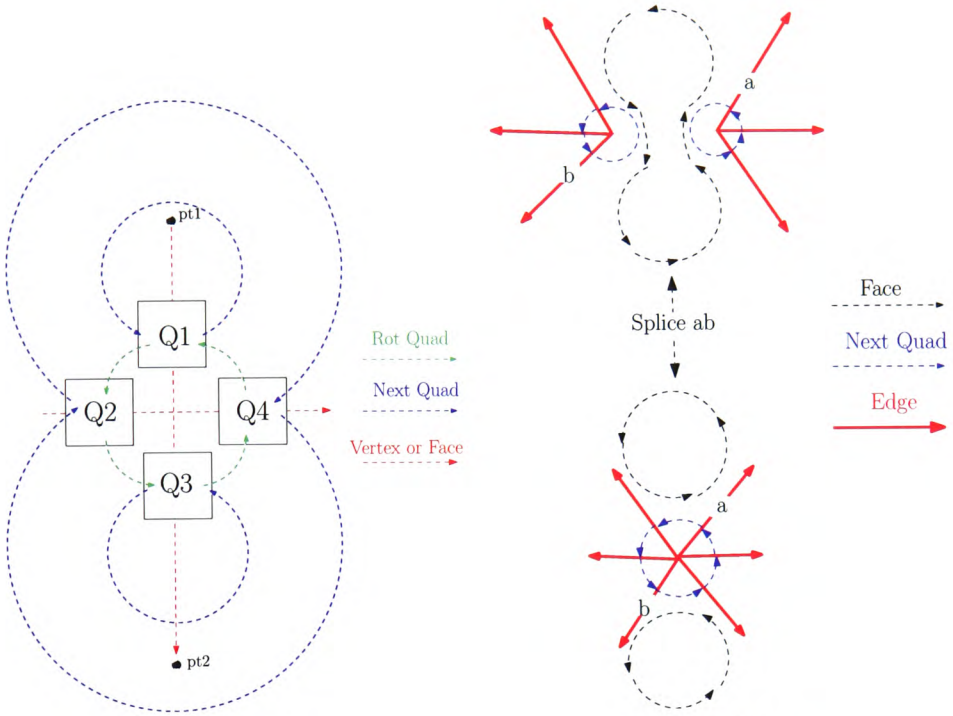
two connected edges disconnects them and merges the faces between them. With Make-Edge and Splice one can delete an edge from a graph and switch the diagonal of a pair of triangles (Guibas and Stolfi, 1985).

To reduce the complexity, "e1" is used to represent the red edge in Fig. 3.5 which the direction of the edge is from "p1" to "p2" (from left to right). "e3" represents the opposit direction of the same edge which creates from "p2" to "p1". "e1" and "e3" are different quads in the same edge which represent different orientation.

3.3.2 The Euler Operators

According to Baumgart (1972); Mäntyla (1986) five operators are enough to modify any arbitrary surface. A continuous surface is used to represent the terrain model; therefore no hole on an individual face (loop/ ring) will be considered. Five normal vectors are used in the model; therefore four operators are enough (Tse and Gold, 2001). The five normal vectors are the number of faces, vertices, bodies (shells), edges and holes (tunnels or bridges) (Tse, 2003). All Euler Operators have corresponding inverse operators that undo the effect of the "positive" operators (Mäntyla, 1981). One more operator is used for building extrusion to increase the efficiency; therefore in total five operators are used, which are:

- "Make Edge Vertex Vertex Face Shell" (MEVVFS) creates a shell, a face and an edge with two vertices to initiate a new model. Its inverse "Kill Edge Vertex Vertex Face Shell" (KEVVFS) destroys the skeletal data structure by MEVVFS leaving an empty model.



(a) Make-Edge (b) Splice

Figure 3.4: Two Topological Operators for the Quad-Edge Structure

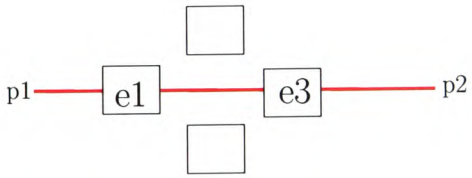


Figure 3.5: An example of an edge

- "Make Edge Face (MEF)" and its inverse "Kill Edge Face (KEF)" are used to subdivide a loop by joining two existing vertices with a new edge. MEF splits one face adding a new edge into the data structure. KEF removes an edge, and two faces merge.
- "Split Edge Make Vertex" (SEMV) splits the input edge and make a new vertex, and its inverse "Join Edge Kill Vertex" (JEKV) joins two edges and kills one vertex.
- "Make Edge Hole Kill Face" (MEHKF) is an additional operator used to extend a TIN model by create a hole or a bridge between two unconnected closed surfaces. Its inverse "Kill Edge Hole Make Face" (KEHMF) removes the hole or bridge by killing an edge and making a face.
- "Make Zero-Length-Edge Vertex" (MZEV) splits a vertex by adding a vertex and a zero-length edge. Its inverse "Kill Zero-Length-Edge Vertex" (KZEV) is used to merge two vertices by killing the zero-length edge and the vertex. MZEV can be replaced by using MEF, KEF and SEMV, but it is much simpler to use MZEV directly.

The Triangulation Operators

The highest level operators used to construct the simple TIN model are "Big Triangle", "Insert Point", "Delete Point" and "Flip" (Swap). The following paragraphs will show detail of the four triangulation operators.

"Big Triangle" The incremental algorithm (Guibas et al., 1992) starts with a big triangle, the initial triangle, which is big enough to contain all the data points. Creation of the big triangle is the simplest way to begin the TIN model. It initiates the triangulation model by creating three vertices and three connected edges with a triangular face.

"Insert Point" When a new data point is inserted, the first step is to locate the triangle in the triangulation containing the point to be inserted. Then, three edges are created to connect the point (pt) to the three vertices of the located triangle, and two faces are created in Fig. 3.11(b). The Walk function (Gold et al., 1977) is used to locate the triangle containing the point to be inserted.

"Delete Point" Deleting a point in the Delaunay Triangulation can be viewed as the inverse of inserting a point. If point "pt" is going to be removed, all the edges connected to "pt", will be deleted. The delete algorithm has to remove all triangles connected to point "pt". Devillers (1999) described the details of the delete algorithm in the Delaunay Triangulation. Delete Point is the inverse operator of Insert Point.

"Flip22" or "Swap" Each triangle forms a circum-circle with its three vertices. According to the Delaunay Triangulation criterion (Delaunay, 1934) no extra vertices may fall within this circle. If there is a triangle to form a circum-circle which has more than three vertices

inside, the "Flip22" operation will be performed. It flips the triangle edges until no circum-circle contains more than three vertices. The operator makes sure that the circum-circle formed by the three vertices of the triangle has no extra vertex inside the circle.

"Big Triangle" initiates the triangulation model creating the first big triangle which is big enough to contain all the data points. "Insert Point" adds data points inside the big triangle one by one. "Delete Point" removes points from the big triangle dynamically. "Flip22" swaps or flips a common edge between two triangles in the triangulation. Besides the topological operators used in Delaunay Triangulation, geometrical operators are used to process the computation. The counterclockwise test (CCW) (Gold et al., 1977; Guibas and Stolfi, 1985), In-Circle test (Guibas and Stolfi, 1985) and Power Computation (Devillers, 1999) are the geometrical functions used in the program.

The Implementation of Euler Operators Using the Quad-Edge Data Structure

Euler's formula for the polyhedral, described by Mortenson (1988); Braid et al. (1978), is always fulfilled during the implementation process. Each Euler Operator is checked to ensure the validity of its performance (Tse, 2003). The implementations of five Euler Operators are described which include MEVVFS, MEF, SEMV, MEFKF, and MZEV.

Fig. 3.6 shows the operators of MEVVFS and its inverse KEVVFS. It creates (kills) a shell with a face, an edge (represent by Q1) with two vertices (pt1 and pt2) from an empty model.

MEF and KEF split and merge faces. The two input edges in MEF must be located inside the same polygon. Fig. 3.7 inputs edges "b" and "a" which are in the same polygon. Edge "e" is created and a new polygon is created by MEF. The input edge is checked to see that it is not a dangling arc before using the KEF operator.

SEMV and JEKV are presented in Fig. 3.8. Input edge "e" is split, and an edge "a" and a vertex "pt" are created in SEMV. Input edge "a" joins edges "e" and "a" by killing edge "a" and vertex "pt" in JEKV.

Two unconnected faces (triangles) are selected to create a bridge or a hole using MEHKF. No common edge and no common vertex connect these two faces. An edge and a bridge or hole (tunnel) are created between two faces, one face is created inside the tunnel or outside of the bridge. The two unconnected triangles (faces) are removed. Therefore there is one face lost in the model. An edge is taken from the two selected triangles for the input in Fig. 3.9 (edge "a1" from triangle "A" and edge "b1" from triangle "B").

MZEV splits a vertex by adding a vertex and a zero-length edge. This operator does not add a face to the model. Two edges are input in the MZEV operator, and they must have the same

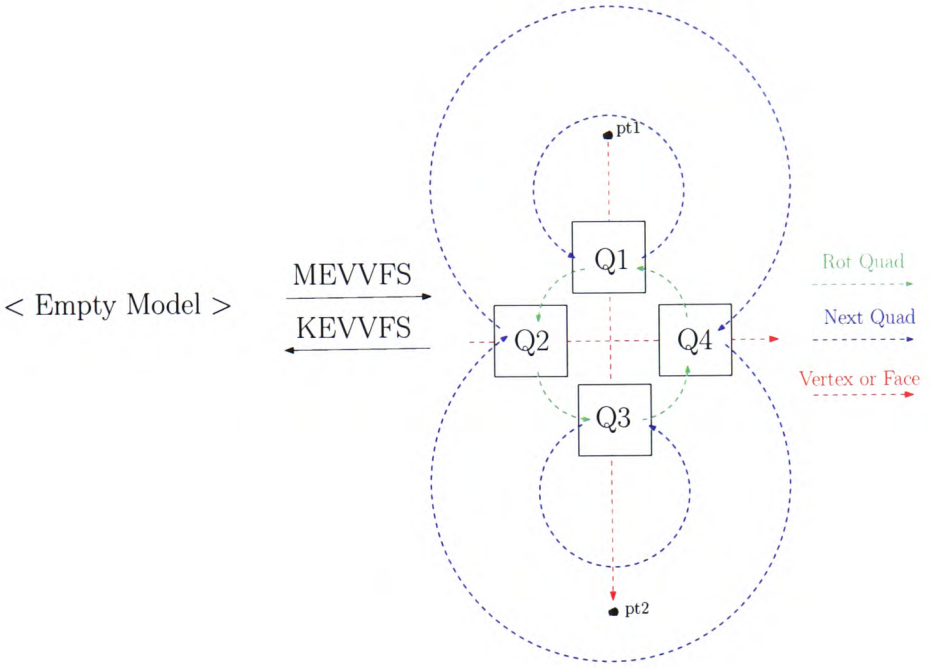


Figure 3.6: MEVVFS ↔ KEVVFS

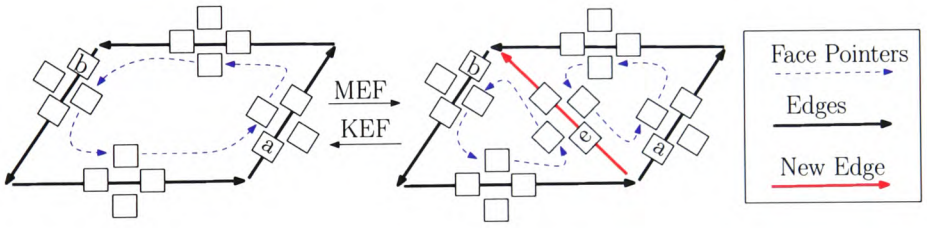


Figure 3.7: MEF ↔ KEF

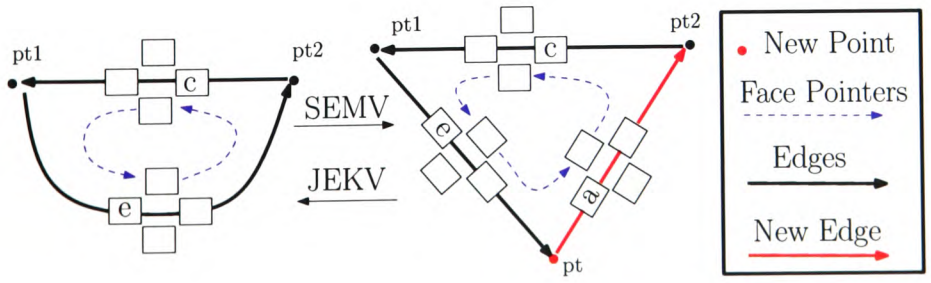


Figure 3.8: SEMV ↔ JEKV

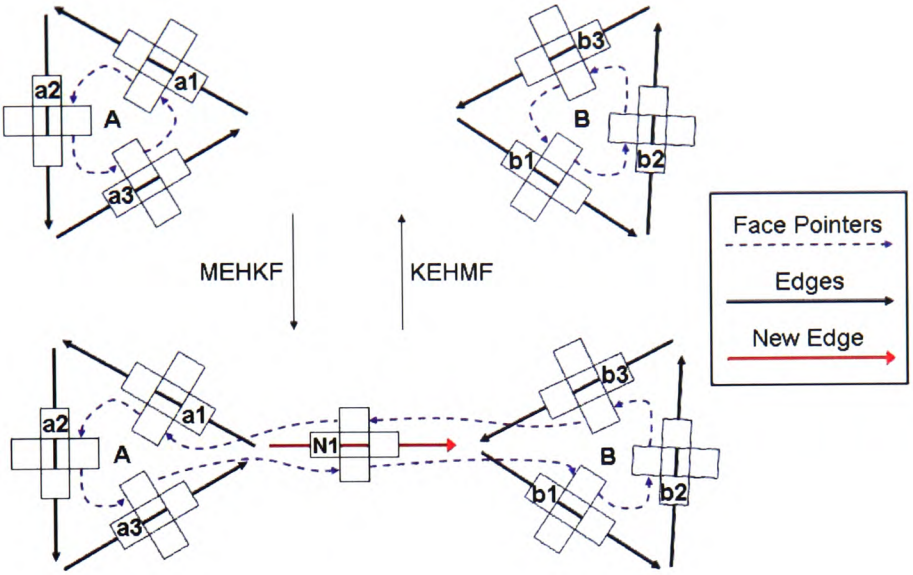


Figure 3.9: MEHKF ↔ KEHMF

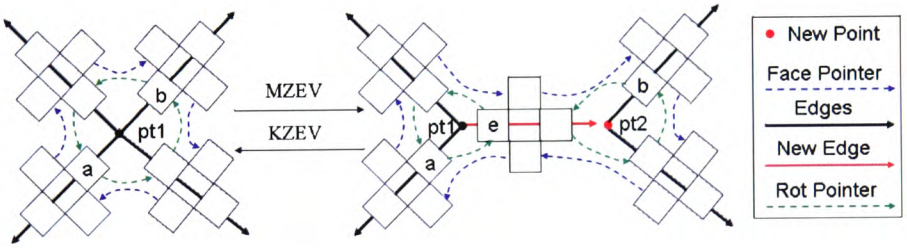


Figure 3.10: MZEV ↔ KZEV

origin. In Fig. 3.10 edges "a" and "b", and a zero-length edge "e" and a new point "pt2" are created.

3.3.3 The Implementation of the Delaunay Triangulation Operators Using the Euler Operators

"Big Triangle" (inverse Kill Big Triangle), "Insert Point" (inverse "Delete Point"), and "Flip22" (inverse "Flip22") are operators of the Delaunay Triangulation. Before performing each operation, a checking procedure is used to check that the input parameters are correct (Tse, 2003).

"Big Triangle" ↔ "Kill Big Triangle" Three Euler Operators, "Make Edge Vertex Vertex Face Shell" (MEVVFS), "Make Edge Face" (MEF), and "Split Edge Make Vertex" (SEMV), are used to create the first big triangle, these operators make sure there is no dangling arc

in the model. Fig. 6.10(a) shows the creation of a big triangle, a shell, an edge ("e1") with two vertices ("pt1" and "pt2") and a face by MEVVFS from an empty model. Then one edge ("e3") and a face are added by using MEF. Finally SEMV splits edge "e3" by adding edge "e2" and vertice " pt3". Three vertices are the input parameters in the function of creating a big triangle. The triangular face runs anti-clockwise.

The inverse operators to kill the big triangle are "Join Edge Kill Vertex" (JEKV), "Kill Edge Face" (KEF) and "Kill Edge Vertex Vertex Face Shell" (KEVVFS) from bottom to top in Fig. 6.10(a). No input parameter is needed to kill the big triangle. Before performing the operation, the triangle must be empty without any data point.

"Insert Point" ↔ "Delete Point" "Insert Point" inserts a new point in an existing TIN model and its inverse "Delete Point" deletes a selected point. The procedure of inserting a point is shown in Fig. 3.11(b) (from top to bottom), and deletion is shown from bottom to top. Point "pt" is input in the triangulation. Before inserting the point, an edge and a face are added using MEF. New edge "N4" is split adding a new edge "N5" at the location of the point "pt" using SEMV. Finally edge "N6" and a face are created using MEF. All the new created edges ("N4", "N5" and "N6") point toward the inserted point "pt". Three steps are used to delete a point from the mesh, if there are only three edges connected to the point to be deleted. Edge "N6" and the associated face are removed using KEF in Fig. 3.11(b). Edge "N4" and "N5" are joined and point "pt" is killed using JEKV. Finally KEF kills edge "N4" and the accompanying face. If there are more than three edges connected to the deleted point, "Flip" the third function of the TIN model will be used. It swaps the diagonal of the quadrilateral.

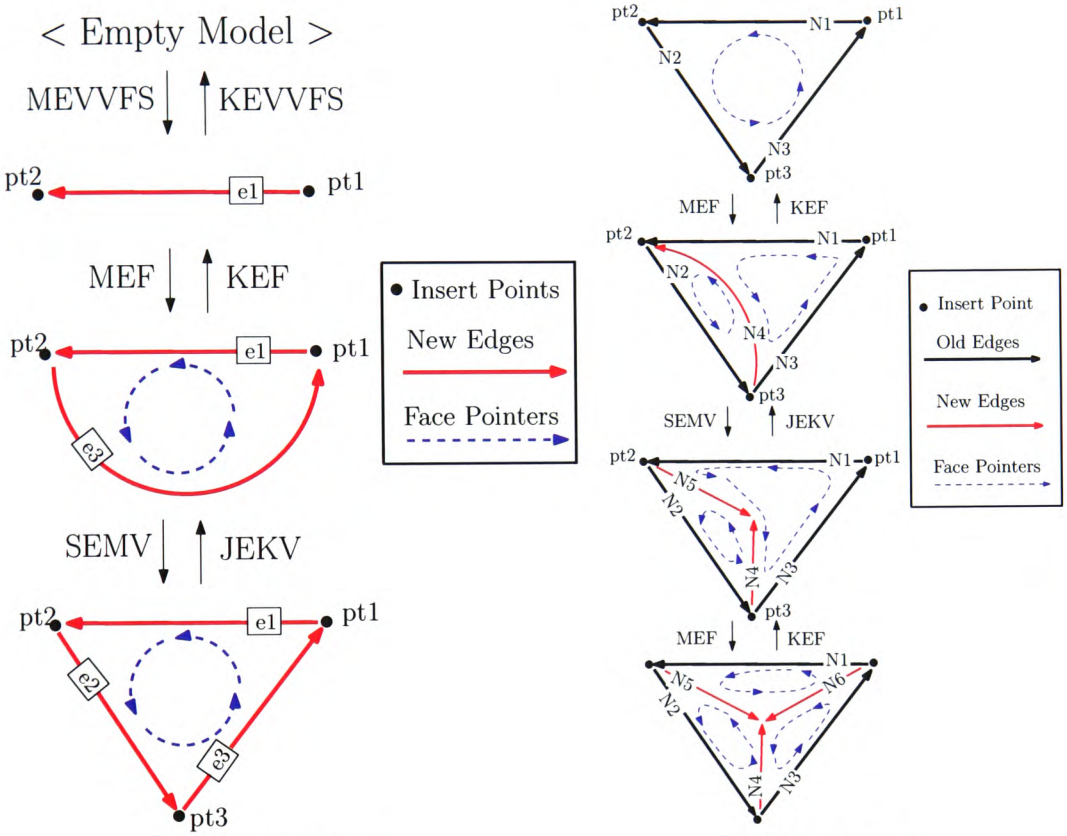
"Flip22" This function is used to check the triangles. First edge "e" is killed using KEF in Fig. 3.11(c); therefore two faces are merged. Then, MEF creates edge "e" and splits the face in two pieces. Edge "e" is the input parameter in the function.

3.3.4 Building Extrusion Using Additional Euler Operators

Extrusion of a building and creating a bridge or a hole (tunnel) are two main ways to extend a TIN model; however the extended TIN model is no longer a Delaunay Triangulation. The methodologies of building extrusion is described in this thesis, but the methodologies of building destruction and creating a bridge or a hole (tunnel) can be found in Tse and Gold (2004, 2002b).

"Make Edge Face" (MEF) and "Make Zero-Length-Edge Vertex" (MZEV) are the only two operators used to extrude buildings. The process of extruding a building is divided into three parts. In each part, only one operator is used, and it is used repeatedly.

To start the process, a triangle or a set of triangles (depending on the shape of the building) and a specified height are input to extrude the building. If a set of triangles is selected, a closed face



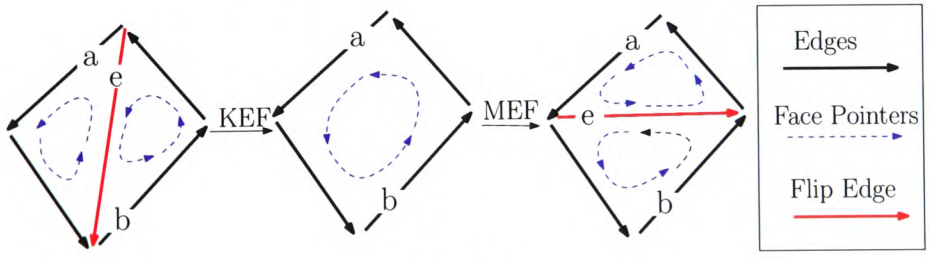


Figure 3.11: Three Triangulation Operators

loop must be ensured before doing the task. To keep the topological connectivity throughout the process, the inverse function of building destruction includes "Kill Edge Face" (KEF) and "Kill Zero-Length-Edge Vertex" (KZEV).

First Step: "MEF" A rectangular building is created in the following example. Fig. 3.12(a) shows two selected triangles with a common edge "e". Their bounding edges are connected as a closed face loop. MEF creates a face and edge "N1", and the direction is from point "pt1" to "pt4"(Fig. 3.12(b)). Three faces are inside the selected rectangle. Three more MEF operators are used to create three more edges and faces in Fig. 3.12(c). Six faces are inside the rectangle.

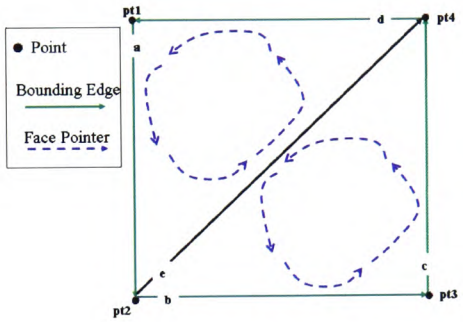
Second Step: "MZEV" Three more MZEV operators are used. If the building has more than four corners, more MZEV operators are used, until the entire buildings' corner points are split. From Figs. 3.13(a) to 3.13(b) four MZEV operators are used to split four points. In Fig. 3.13(a) MZEV splits point "pt1" by adding a point "pt5". Fig. 3.13(b) shows point "pt2" is split by adding point "pt6", and point "pt3" is split by adding point "pt7", and "pt4" is split. The new points are vertically on top of the respective split points. Four edges are created at each corner of the building to link the split and the newly created points, e.g. edge "N5" links points "pt1" and "pt5". Fig. 3.13(c) shows the extruded building with four rectangular face surround it.

Third step: "MEF" The rectangular faces on each side of the extruded building is split into triangles by adding a diagonal edge. An MEF operator is used to add an edge which runs from point "pt7" to point "pt2" and to split the face into triangles in Fig. 3.14(a). Three more MEF operations are performed to split the three faces into triangles. Fig. 3.14(b) shows the extruded building in rectangular shape on a TIN model. All the faces on the building are split into triangles.

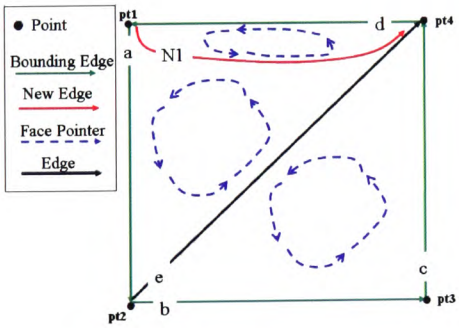
KEF and KZEV are the two major operators used to remove buildings. A set of triangles is the input of the operator. If a point is located inside the triangles, the point needs to be deleted before removing the building. Modelling the building is more important than destruction in this project; detail can be found in (Tse, 2003).

3.4 Chapter Summary

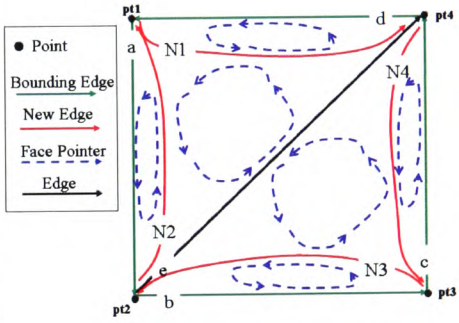
Airborne laser scanning (ALS) captures all the objects on the terrain includeing vegetation, buildings, forestry and so on. There is no additional information provided to describe the captured data; therefore algorithms have been developed to extract useful information for further analysis.



(a) Two selected triangles

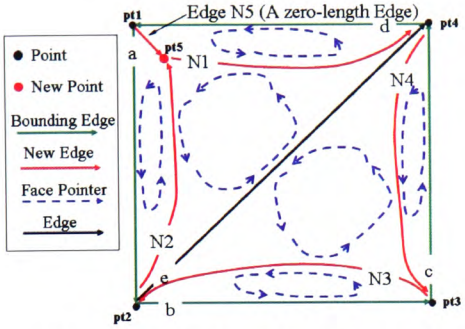


(b) MEF splits a triangle face

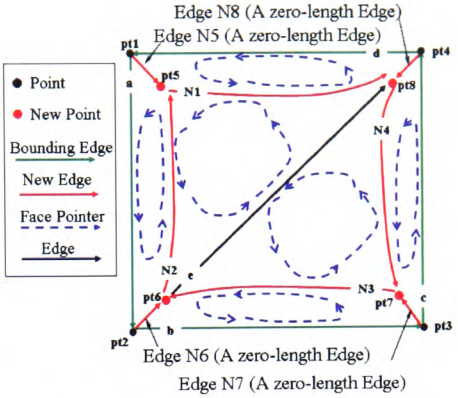


(c) Four more faces after four MEFs

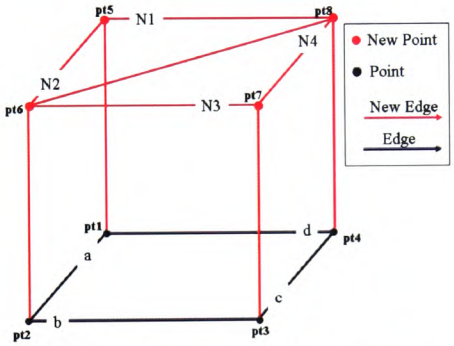
Figure 3.12: First Step(MEF): Building Extrusion



(a) First MZEV to uplift the building

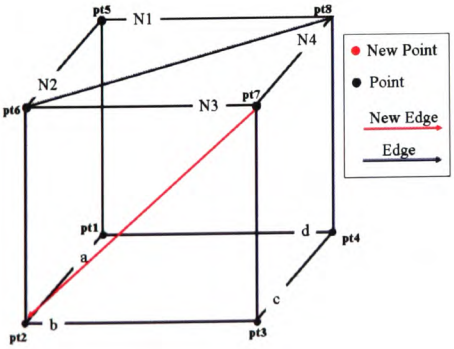


(b) The extruded building after four MZEV

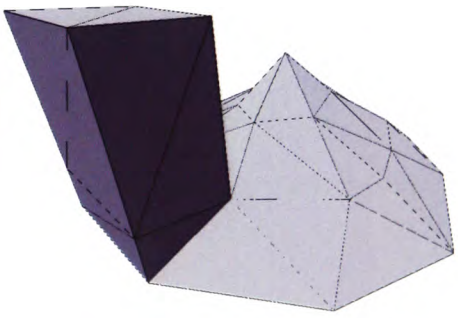


(c) 3D view of the extruded building

Figure 3.13: Second Step (MZEV): Building Extrusion



(a) MEF splits the surrounding rectangular faces to triangles



(b) The final 3D view extruded building

Figure 3.14: Third Step (MEF): Building Extrusion

To use LIDAR for building reconstruction, it is popular to start with removing man-made (e.g. buildings and bridges) and natural (e.g. trees or forestry) non-terrain objects and generating a bare-earth model. Twelve algorithms (e.g. Jacobsen and Lohmann (2003); Elmqvist (2001); Sohn and Dowman (2002); Vosselman and Maas (2001); Roggero (2002)) were reviewed and it was found that most of the filtering algorithms do not return a perfect result. Normally the filtering algorithm is needed because the reconstructed buildings can be pasted on top of the bare-earth model (like playing Lego^{TM}). In this project, buildings are extruded from the terrain surface; therefore no filtering algorithm is needed.

Building extraction looks for the building outline and many approaches combine using additional data sources. Some of them (Rottensteiner, 2001; Khoshelham, 2005; Sohn and Dowman, 2001) used aerial images, and some of them used two dimensional ground plans (Brenner and Haala, 1998; Hofmann et al., 2003; Vosselman and Dijkman, 2001) to search for the building boundaries. If the outline of roof is used as the building boundaries, based on a single calculation may decrease the accuracy of the result. The approach of this research will not use any additional data source to extract the building footprint, and then compares the findings to the roof outline to improve the result.

A model-driven approach (Brunn and Weidner, 1997a; Bonn, 1999; Vosselman et al., 2004) is commonly used to reconstruct building models which searches for the best model from the library to fit the building outline. However it takes time to create the library and it is not flexible for the building shape. A data-driven approach (Wang and Schenk, 2000; Hofmann et al., 2003; Maas and Vosselman, 1999) works better because it allows different roof patterns (e.g. multiple level flat or peaked roofs). The flexibility of the building shape is an important issue for building reconstruction; therefore the data-driven approach is used in this project.

The objective of this project is to use raw LIDAR data to reconstruct a building model without additional data sources and pre-defined building models while preserving the topological connectivity; therefore an alternative reconstruction method is found. The following chapters will shows the methodology of using a different approach for building reconstruction from raw LIDAR data.

Chapter 4

Building Outline or Building Block Recognition

The technical background gives in the first section of this chapter describes the different technical terms which will be used in this chapter. Section two gives a brief introduction of the chapter. The Mastermap Topographic Layer from the Ordnance Survey is used to recognise the building footprint in the third section. Alternate ways of building block recognition are shown in Sections 4.3 and 4.4. The final section gives a chapter summary.

4.1 Technical Background

The technical terms use in this chapter which are 'Delaunay Triangulation', 'Principle Components Analysis (PCA)', and 'Minimum Spanning Tree (MST)'.

4.1.1 Delaunay Triangulation and its Dual Voronoi Diagram

The Delaunay Triangulation (DT) introduced by Delaunay (1934) is a commonly used triangulation method for generating an unstructured mesh. Gold and Snoeyink (2001); Gold (1979) agree that the TIN model can be improved by using the Delaunay Triangulation. Fig. 4.1 shows triangles A and B; triangle A formed by points p1, p2 and p3, triangle B by point p2, p3 and p4. The properties of Delaunay Triangulation are (refer to Fig. 4.1):

- Each polygon edge is the straight-line dual of the edge of a Voronoi Diagram (VD). Edge p2 to p3 is the straight-line dual of the Voronoi edge v1 to v2 in the figure.
- Each face is a triangle with three points that are cocircular. The vertices of each triangle form a circle which contains no other points inside its circle. For example vertices p1, p2, and p3 of the DT form circle e1 which contains no other points inside the circle.
- Each face (triangle) of the DT corresponds to a vertex of VD.
- Each edge of the DT corresponds to an edge of VD.

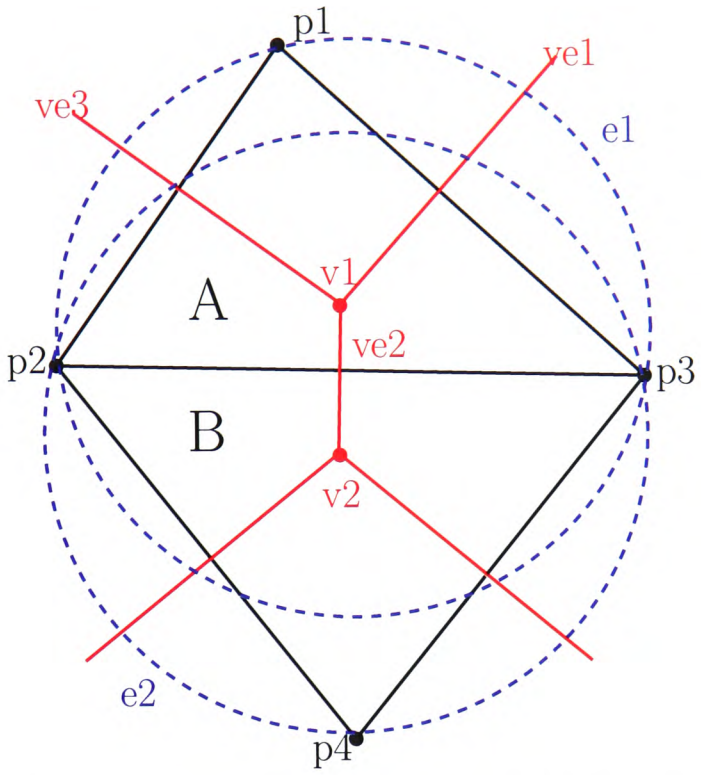


Figure 4.1: Two Triangles in a Delaunay Triangulation

- Each point (node or vertex) of the DT corresponds to a region (polygon) of VD.
- The boundary of the Delaunay network is a convex hull.

The following are the properties of Voronoi Diagram (refer to Fig. 4.1):

- Each edge is the straight-line perpendicular to the edge of a DT. Edge v1 to v2 is the straight-line perpendicular to the DT edge p2 to p3.
- Each polygon is convex (red polygon in Fig. 4.2).
- Each vertex is the centre of the circum-circle of its corresponding triangle. Vertex v2 of the VD is the centre of circum-circle e2 of its corresponding triangle (p2, p3 and p4).
- Each vertex connects to three edges (vertex v1 connects to edges ve1, ve2 and ve3).
- Each cell of the VD corresponds to a vertex of DT.
- Each edge of the VD corresponds to an edge of DT.
- Each vertex of the VD corresponds to a triangle of DT.

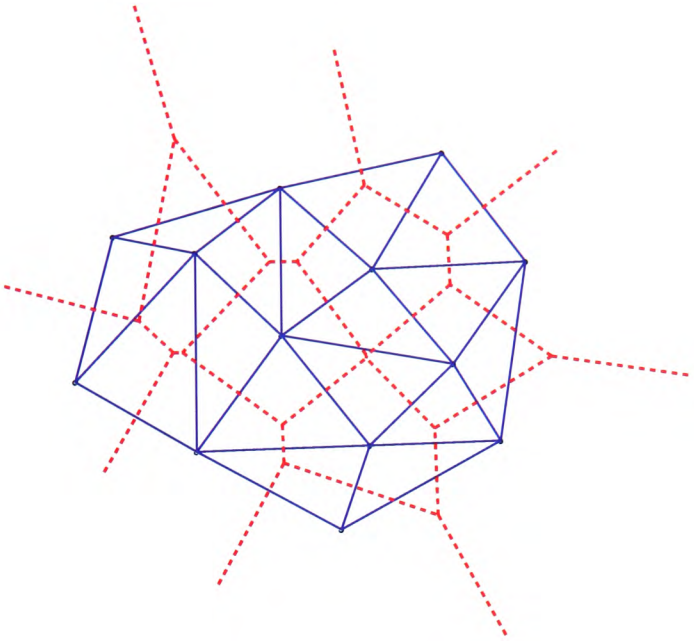


Figure 4.2: The duality relationship between the Voronoi tessellation and Delaunay Triangulation

4.1.2 Principle Components Analysis (PCA), Eigenvalue and Eigenvector

PCA is a classical statistical method, and which is also called the Karhunen-Love transform or the Hotelling transform. The objectives of PCA are to discover or to reduce the dimensionality of the data set, and to identify new meaningful underlying variables. PCA is mostly used as a tool in exploratory data analysis and for making predictive models. The results of a PCA are usually discussed in terms of component scores and loadings.

The mathematical way to define PCA is an orthogonal linear transformation which transforms the data to a new coordinate system; the largest variance by any projection of the data comes to lie on the first coordinate (the first principal component), the second largest variance on the second coordinate, etc. Theoretically this is an optimum transform for a given data in least squares terms.

PCA is based on eigenanalysis of the correlation or covariance matrix. The eigenvector associated with the greatest eigenvalue has the same direction as the first principal component. The eigenvector has the second greatest eigenvalue gives the direction of the second principal component.

PCA is used in the covariance method to project 3 dimensional data onto a 2 dimensional subspace in this project. The lowest eigenvalue and its associated eigenvector are used to find the lowest variance on the data.

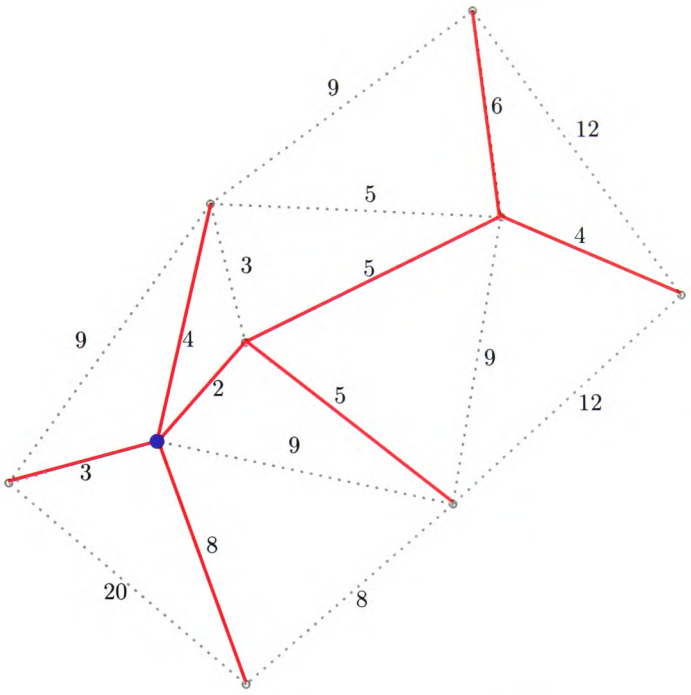


Figure 4.3: An example of a MST

4.1.3 Minimum Spanning Tree (MST)

An MST is a graph, subgraph or tree which is connected and undirected, but all the vertices are connected together. A single graph can have many different spanning trees. Weight can be assigned to each edge, and the weight can be a number representing how unfavourable it is. An MST is formed by searching the least total assigned weights of a spanning tree. Fig. 4.3 shows an example of an MST. The red line is the MST of a planar graph (the gray dotted line), and the numbers are the weights of the edges. The blue point is the starting point of the MST in Fig. 4.3, and the red graph (or tree) shows all the minimum weighted path to all the destinations (points).

Fig. 4.4 shows an example which a salesman needs to go from Point A (the blue starting point in Fig. 4.3) to Point B. If there are two possible ways, the salesman has to choose the best (lowest weight) path by adding up the weight of each path. The total weight of Blue and Green paths are $2 + 5 + 4 = 11$, and $9 + 12 = 21$ respectively. Though the Green path seems to have shortest connection to the destination, the best way for the salesman goes from Point A to Point B is the Blue path with the minimum weight.

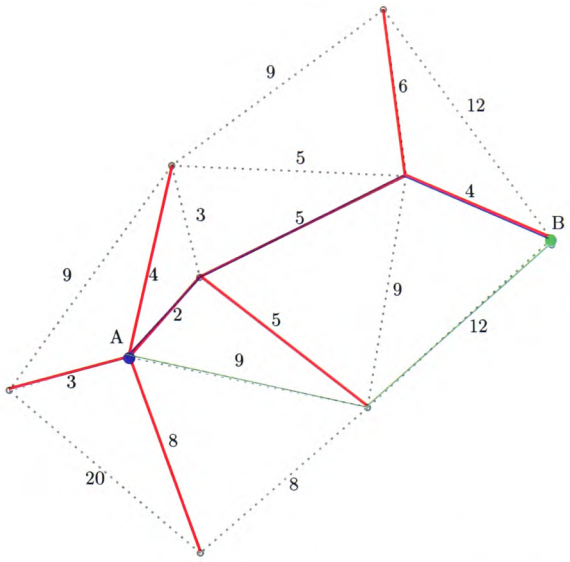


Figure 4.4: The shortest path from Point A to Point B

4.2 Introduction

The building outline (boundary) separates a building block (usually represented by points with larger z-values) and its surrounding terrain surface (represented by lower z-value points). Raw LIDAR data contains all the high and low points and it is not easy to automatically recognise the building boundaries. This chapter shows two methods of tracing building boundaries from LIDAR data.

In the past, many researchers tried to combine different data sources (e.g. ground plans or photogrammetric data) to find out building boundaries from raw LIDAR data. However it is necessary to discover a method which uses LIDAR data only, especially when other data sources are not available. A method that separates building blocks (high land) from the terrain (low land) is described in section one, a "Statistical-Testing" model.

A "Voronoi Mosaic City Model" will be described. Though it still needs further development to improve the accuracy of the result, it gives an alternative method to extract or locate the building boundaries.

The use of existing 2D map data is fairly common to reconstruct the 3D city model because it provides an easy and quick solution. In this project the Mastermap Topograhic Layer is provided by the Ordnance Survey in the UK. The details of using the existing 2D map data will be shown in the third section.

4.3 Statistical-Testing Method

If the high data points represent building blocks, then the terrain surface is represented by the lower points. A Statistical-Testing method is used to separate the high and low area by searching for the vertical segments or sharp 'edges' in vertical terms. Then the vertical wall segments are connected to form a closed boundary which is the building block. Sections 4.3.1 to 4.3.5 describe the method. Then the procedures of the method are listed in Section 4.3.6. The last section shows the advantages and disadvantages of using the method. Artificial data are used to illustrate the method. Validating the approach using real data is shown in Chapter 7.

4.3.1 Data Points Resampling

Several procedures are involved in the Statistical-Testing method. The first step is to create a Delaunay Triangulation using the raw LIDAR data points. Then the high density data points are sampled to a lower resolution and another triangulation is created using the sampled points. The sampling method creates a lower resolution triangulation and its dual Voronoi diagram as an index layer. The Voronoi cells are much bigger than those from the original data. Some big Voronoi cells may contain low and high data points which may contain part of a building boundary. The aim of this method is to search and locate the vertical segments in the Voronoi cells.

The first step of creating a lower resolution Voronoi diagram is to set a disc circle, for example the threshold circle in Fig. 4.5. Then the raw LIDAR data are read one by one. A point is accepted if the distance between the first and the second data point is bigger or equal to the diameter of the threshold circle. In Fig. 4.6 Point B locates inside the threshold circle of Point A; therefore it is rejected. However Point C is outside the threshold circle of Point A, is accepted. Fig. 4.7 shows first four accepted points. The search is continue, until it finds another point that is outside the threshold circle (diameter). Then the point will be accepted.

The threshold size used to create the Voronoi index layer which can be affected by the resolution of the input data points. If big Voronoi cells (index layer) contain too less data points, it is not enough to find the difference between the low and high points. However the big Voronoi cells contain too much data points, it is not accurate to locate the vertical wall between the high and low points. 0.6 - 1.2 meter is the threshold size normally used in this project

Fig. 4.8 shows the raw data points from LIDAR. The red points are the accepted points. Each accepted point is separated from the others by at least a diameter of the threshold circle distance.

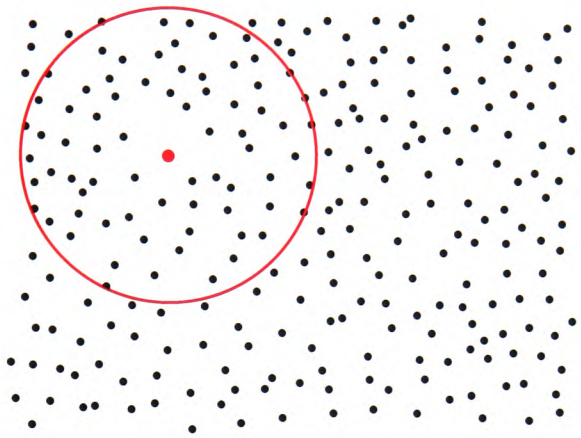


Figure 4.5: The red circle is the threshold circle with the red point (accepted point) as the centre point

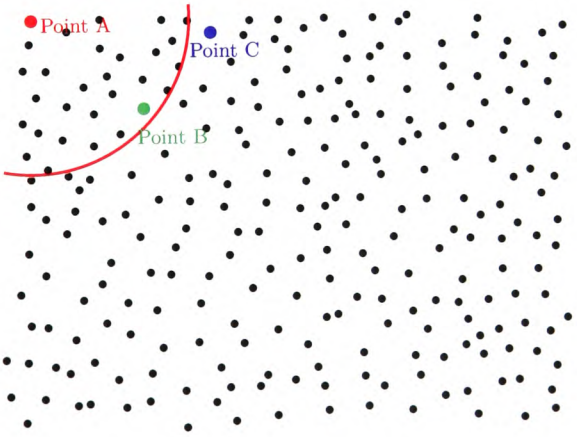


Figure 4.6: Point B is inside the circle and Point C is outside the circle

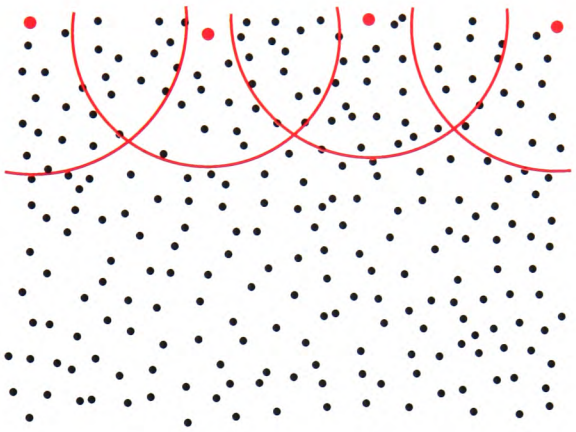


Figure 4.7: Four accepted points and the searching will continue row by row

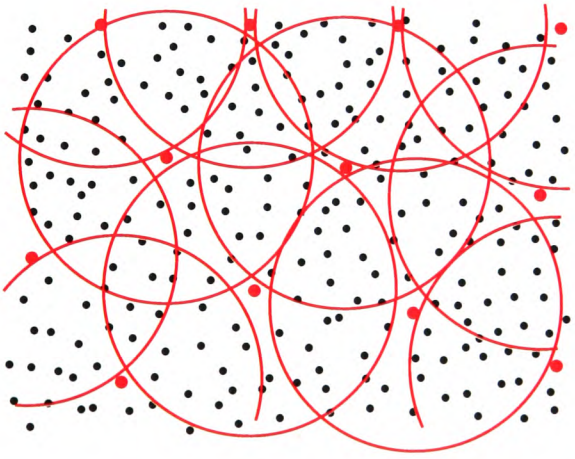


Figure 4.8: The red points are the accepted points and red circles are the threshold circles

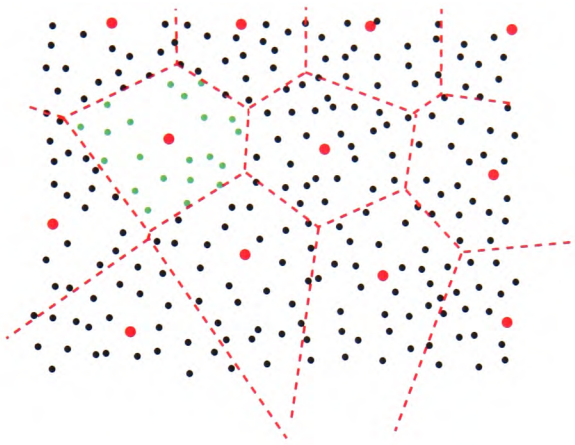


Figure 4.9: The green points are extracted for comparison

4.3.2 Wall Segments Searching

The sample points are used to create a second Delaunay Triangulation (TIN model) and its dual graph, Voronoi Diagram, is used. Raw data points (blue points in one big Voronoi cell (Fig. 4.9) inside each big Voronoi cell are extracted for comparison. Several steps are used to find the line which separates the high from the low points in the Voronoi cell.

All the raw LIDAR data points (the blue high density points in Fig. 4.10) in each big Voronoi cell are extracted. Then PCA is used to analyse the extracted data points, the calculated result includes three eigenvectors and the associated eigenvalues. The eigenvector with the smallest eigenvalue indicates the line of sight (fold axis) (Charlesworth et al., 1975) between the high and low points. Fig. 4.11 shows the eigenvector in 2D and 3D view in a big Voronoi cell. The blue points are the low elevation points and the red points are the high one. The direction of the wall segment is found and the next step is to look for the position of the segment.

The location of the wall segment is found by trial and error. A line with the calculated direction is put at one end of the big Voronoi cell; therefore the line separates the data points into two groups. An averaged height is calculated in each group. Then the line is shifted slightly apart (10 cm) from its original place. Two groups of data points are slightly different because of moving the line, and the average heights are re-calculated. The line is shifted continuously until it reaches the other end of the big Voronoi cell. Fig. 4.12 shows the green lines which are the guess lines for the best location. The next step is to chose which green line separates the data points to give the best location.

The best location is the line which separates data points into two groups and the difference between the averaged elevations is the largest. Each pair of the mean elevations is compared

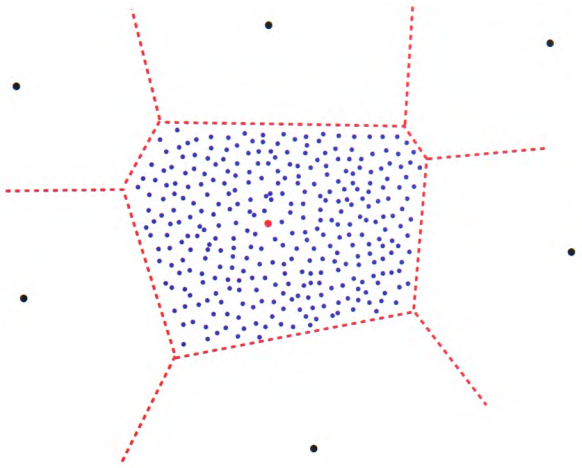


Figure 4.10: LIDAR data points in the big Voronoi Cell

and the difference is calculated. Fig. 4.13 shows a green line which has the biggest difference of the height values (the high (yellow) and the low (blue) points).

Then the chosen line is used to split the big Voronoi cell and forms the wall segment of the building. One Voronoi point is added to split the cell. The green point is the new inserted point in Fig. 4.14 and the distances between the line and the green and red points are the same. If a line is drawn from the green to red points in Fig. 4.14, the drawn line is perpendicular to the chosen (red) line.

The next step is to update the z-value of the new inserted and original Voronoi points in the big cell (the red and green points in Fig. 4.14). The data points of the two groups (the yellow and the blue points) are extracted to calculate their averaged heights. Then the z-value of the big Voronoi cells are updated with the averaged heights. The final step is to store the split edges in a list.

4.3.3 Wall Segment Tracing

If the found wall segments are able to connect and form a closed loop, it indicates a building block exists. All the wall segments, separating the high and low points, are connected in Fig. 4.15. The first split Voronoi edge is picked from the list which stored all the split edges in Section 4.3.2. Each edge connects to two other edges and the other two edges are searched in an anti-closewise order to find the next wall segment. Figs. 4.16 to 4.18 are used to illustrate the tracing procedures. A threshold is set to determine whether the difference between two z-values of the two Voronoi points is big enough to form a wall segment.

The steps of searching the closed loop are:

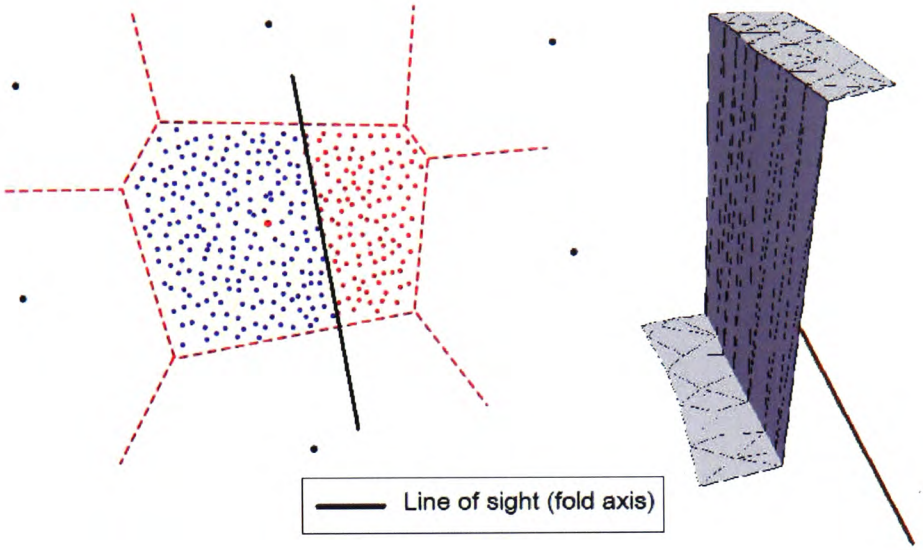


Figure 4.11: The 2D and 3D of the eigenvector with the smallest eigenvalue

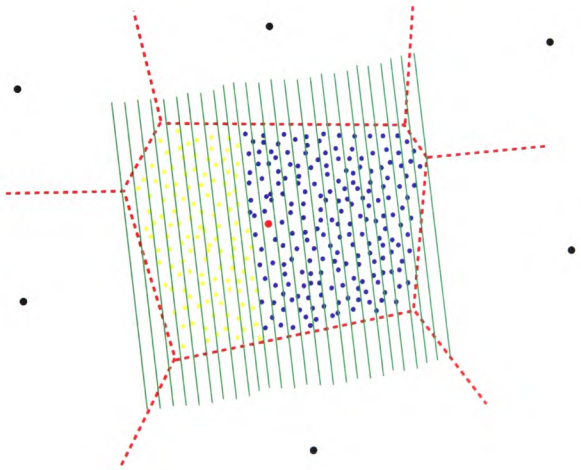


Figure 4.12: Move the lines (green lines) in a big Voronoi cell

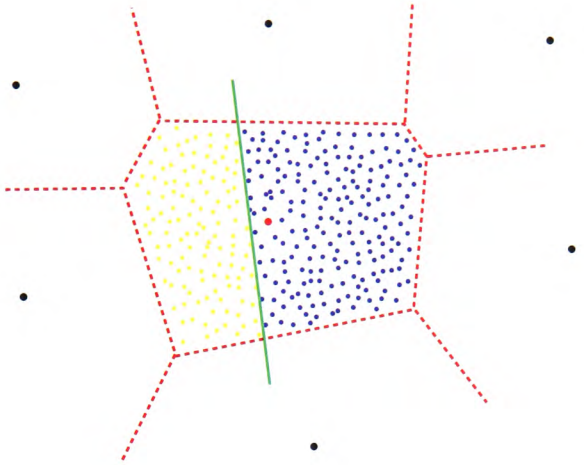


Figure 4.13: The line which separates the low and high points

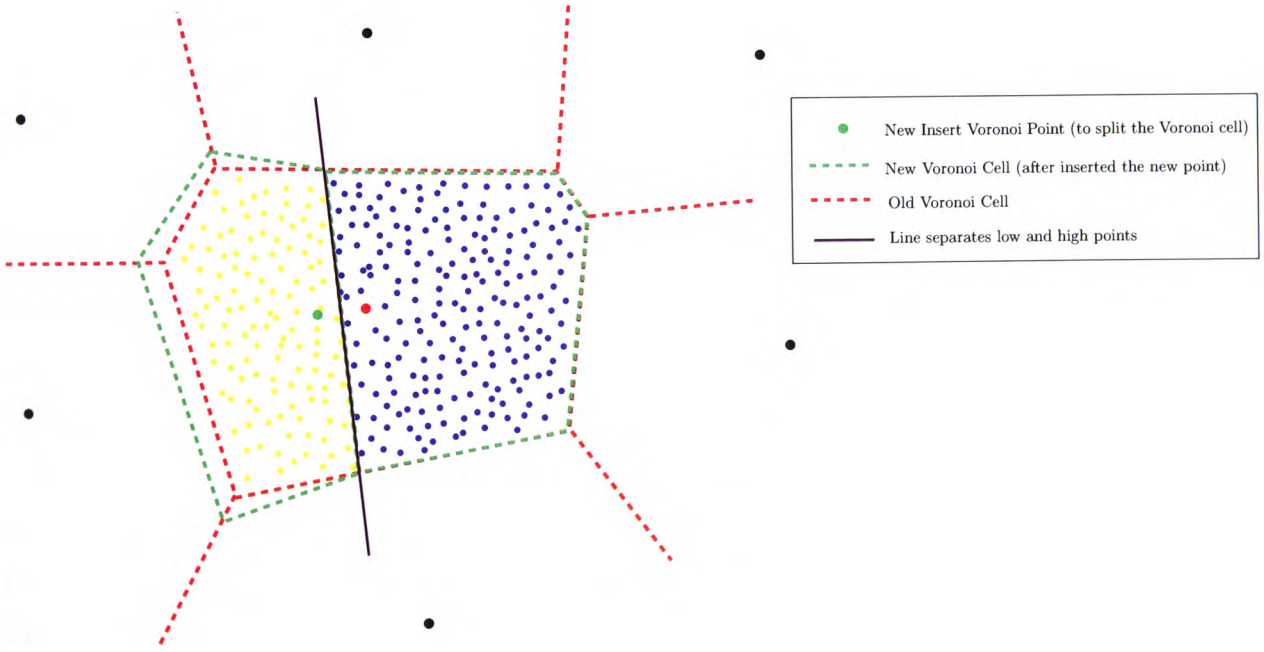
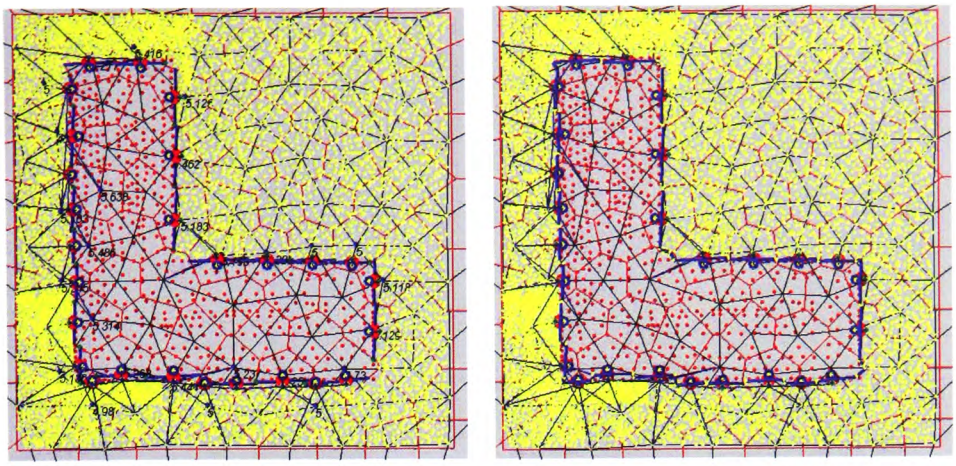


Figure 4.14: Add one point to split a cell



(a) Averaged z-values of the sample Voronoi cells

(b) Connects all the Vertical Segments

Figure 4.15: Searching a closed loop of the vertical segments

1. Pick up the black solid line 1 which is the first split Voronoi edge.
2. Navigate to the immediate next Voronoi edge in anti-clockwise order (the blue dashed line 2 in Fig. 4.16).
3. Compare the elevation of the Voronoi centre points A and B.

   a)connect them if the elevation difference is larger than the threshold.

   b)do nothing if the elevation difference is smaller than the threshold.
4. Keep checking each adjacent Voronoi edge until the checking edge returns to the black line 1.
5. Pick up the last connected Voronoi edge (the green line) and keep searching other edges.
6. Stop checking when the checking edge returns to the first split Voronoi edge (the black line 1).

   a)A building exists if the checking edge returns to the first split Voronoi edge.

   b)No building is found if the connection stops.

Fig. 4.16 shows an example of tracing the wall segments. Edge 1 is the first split edge on the list, and Edge 2 is the next edge in anti-clockwise direction. The difference of the z-values between points A and B is greater than the threshold.

The blue dashed line is one of the wall segments because point A is much higher than point B in Fig. 4.16. However the green dashed line is not the wall segment, the elevation difference between points C and B is not bigger than the threshold.

The blue dashed line is confirmed as a part of the wall segment in Fig. 4.16, the next step is to search the next wall segment connecting to it. The green dashed line 4 is the immediate next

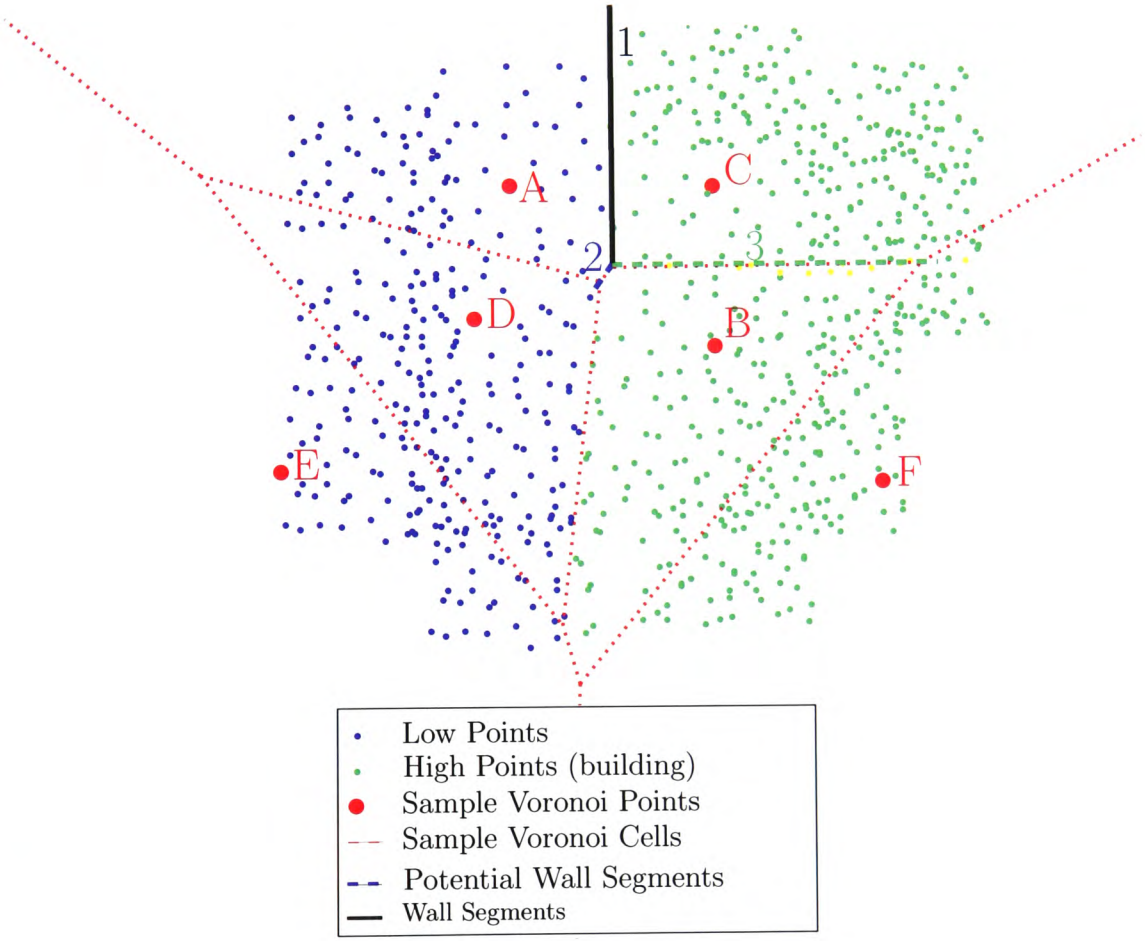


Figure 4.16: The black line is the first split Voronoi edge

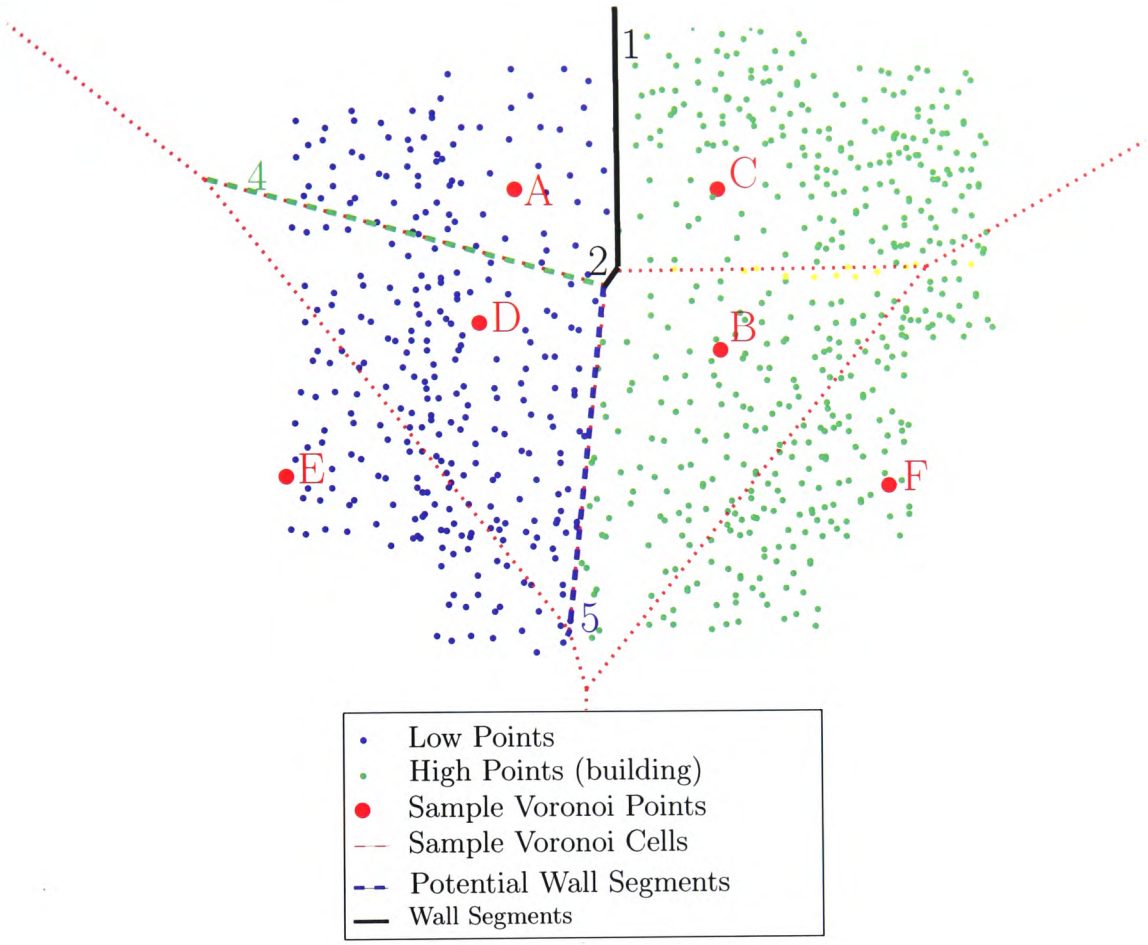


Figure 4.17: Either the green or blue lines is the wall segment

Voronoi edge in anti-clockwise order of the black line 2 in Fig. 4.17, and the green line separates Points A and D. The elevation difference between Points A and D is smaller than the threshold; therefore the green dashed line 4 is not a wall segment.

The blue dashed line 5 separates the high point D and the low point B; therefore it becomes one of the wall segments (Fig. 4.17). Fig. 4.18 shows the black lines separate the high and low points.

The black lines are part of the wall segments of a building. The searching continues until it returns to the first split Voronoi edge and a closed loop is formed. When a closed loop is found, it shows a building exists as in Fig. 4.15(b).

An extension of a building is a vertical wall separating the building into two parts. When there are extensions to the main building, the wall segments tracing task is performed twice. The searching procedures are the same, but the direction is in a clockwise order. Then two sets of

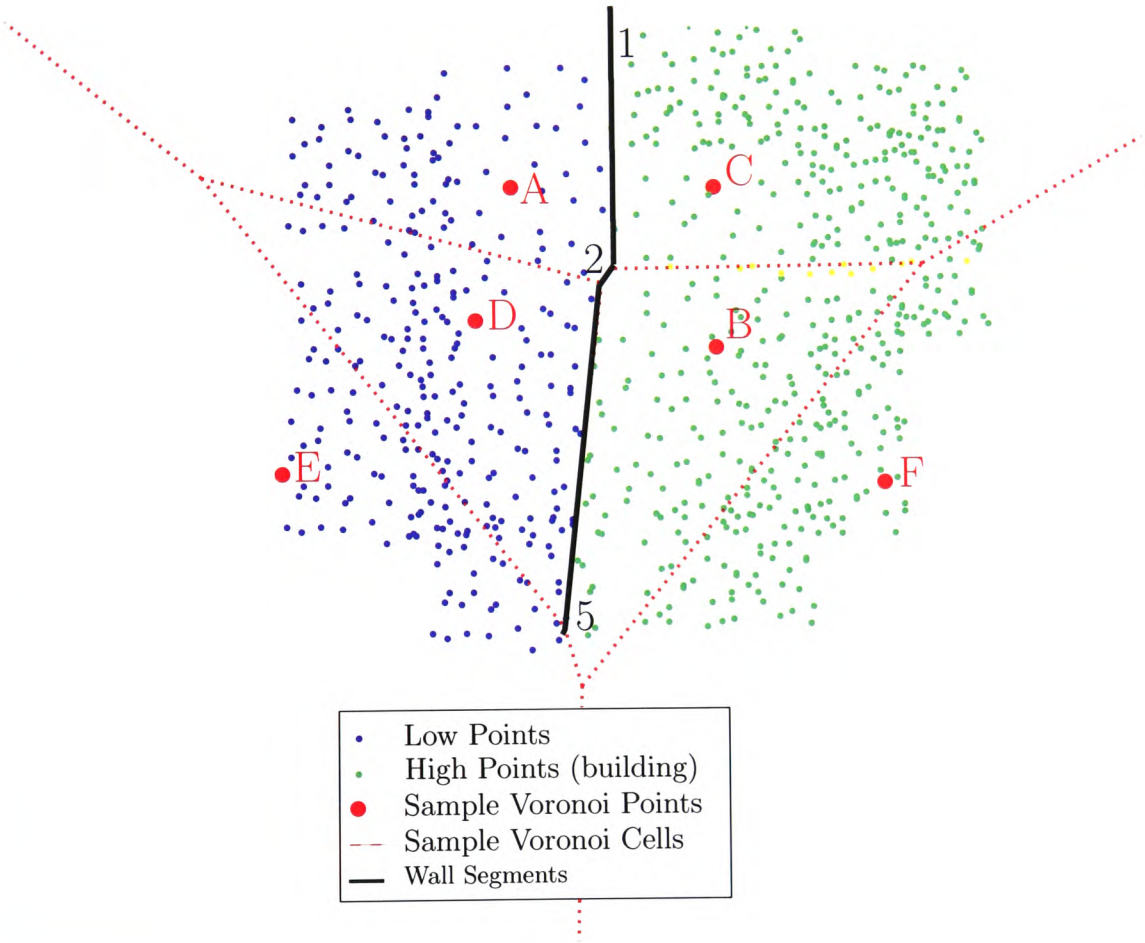
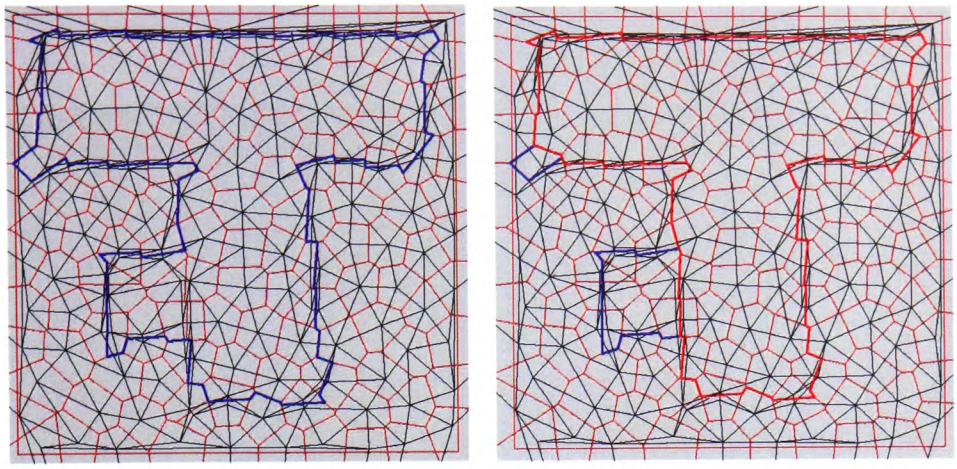


Figure 4.18: The black lines are the wall segments which separate the high and low points



(a) Vertical segments of the whole building with the extension

(b) Vertical Segments of the main building without the extension

Figure 4.19: The found vertical segments of the building

vertical segments produce two sets of building corners. The next step is to compare the two sets of building corners and find the building boundaries with its extensions. A T-shaped building with an extension on one side is used to illustration the steps.

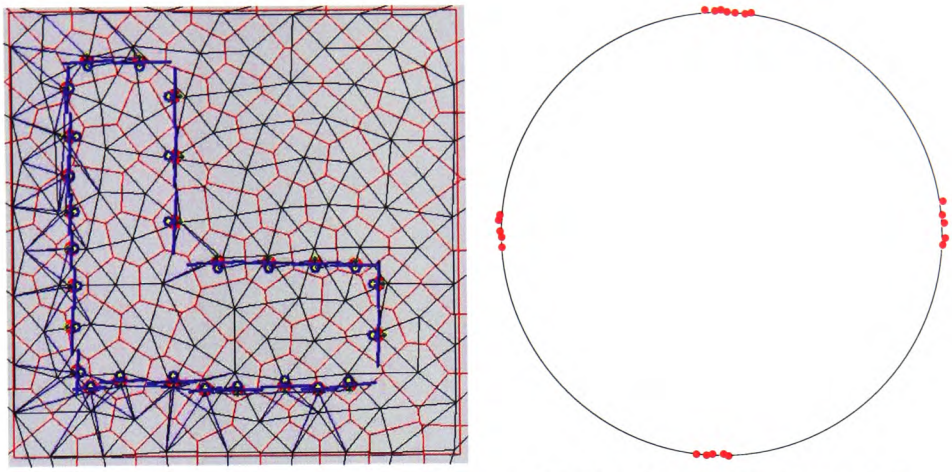
Fig. 4.19(a) shows the connected vertical segments which were found in anti-clockwise direction. It gives a closed loop including the building extension. Fig. 4.19(b) connects all the vertical segments in red to form a closed loop in a clockwise order. The segments enclose the main building only.

It will continue to find the building block if a closed loop is found. Further calculation is needed to sharpen and connected the split Voronoi edges to form the building outline and find the building corners. The split Voronoi edges are clustered according to their orientation and geographical location.

4.3.4 Wall Segment Clustering

To cluster the Voronoi edges according to orientation, the split Voronoi edges (vectors) are plotted on a circle (Fig. 4.20). MST is used to cluster the split edges on the circle. Figs. 4.21 and 4.22 show the clustered vectors on the circle in different colours and the clustered Voronoi edges.

Each group of the edges (same orientation) may represent two different sections of the building outline (Fig. 4.22). Therefore they are clustered according to their geographical location. The clustered edges are separated if they are not close to each other. Finally each group of the clustered Voronoi edges represents each section of the building outline (Fig. 4.23).



(a) The blue lines are the split Voronoi edges (b) The edges are plotted on a circle

Figure 4.20: The split Voronoi edges and plot vectors on a circle (right)

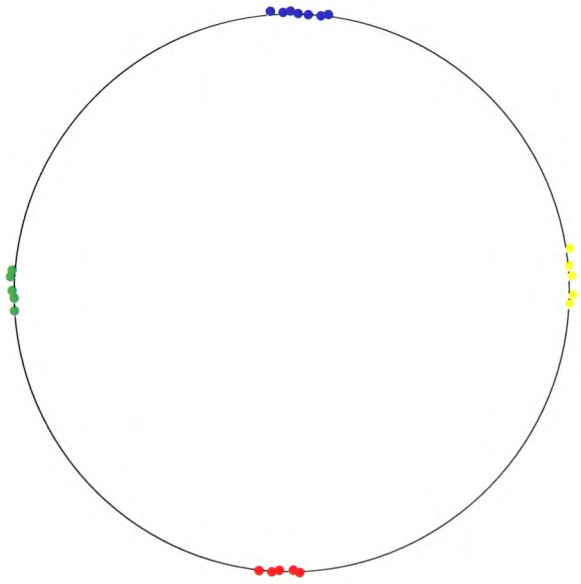


Figure 4.21: The clustered vectors on the circle

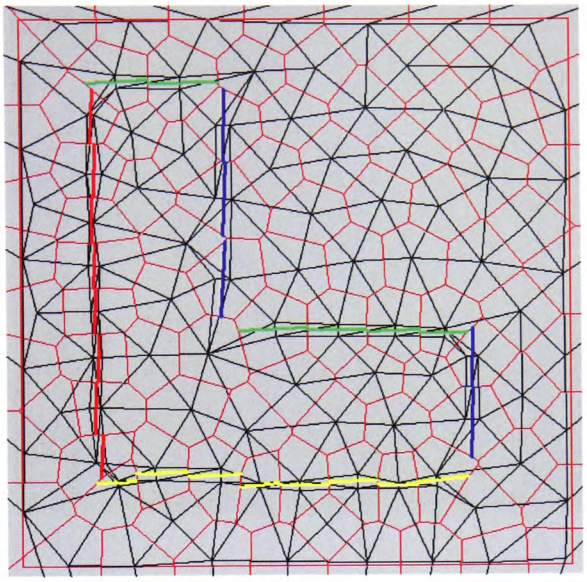


Figure 4.22: The green and blue edges represent two different sections on a building outline but clustered in the same group

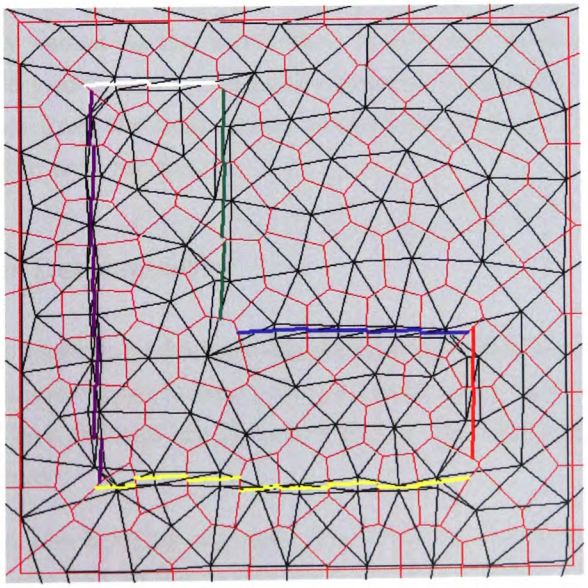


Figure 4.23: Each colour of the clustered edges represents each section of the building outline

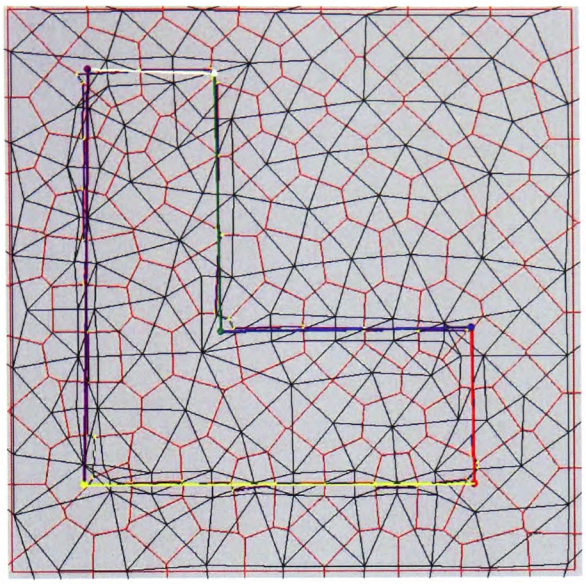


Figure 4.24: Six best fit lines calculated from the six groups of split edges

Each group of the split Voronoi edges share a similar orientation and geographical location. The split Voronoi edges act as vertical walls of the building to separate the high and low points. Fig. 4.23 shows six groups of split edges in six different colours. Then a regression method is used to find the best fit line in each group of the edges. The ordinary linear regression cannot be used because it minimises the sum of the squared vertical distances between y data values and the corresponding y values on the fitted line. It does not work when the line is parallel or almost parallel to the y-axis. An orthogonal regression is used to minimise the orthogonal (perpendicular) distances from the data points to the best fit line. The orthogonal regression method is described in the next section.

4.3.5 Building Corner Determination

The orthogonal regression method calculates the eigenvector which has the biggest eigenvalue in each edge group. Then the average (middle point) is found and a best fit line is formed for each group. In Fig. 4.24 six best fit lines are formed with the clustered edges and their eigenvector (for the biggest eigenvalues).

The next step is to find the building corners. The order of the split edges have been found already and the order of the best fit line is the same as the split edges. A line to line intersection method is used to calculate the intersection points between two adjacent best fit lines. Fig. 4.25 shows the building outline.

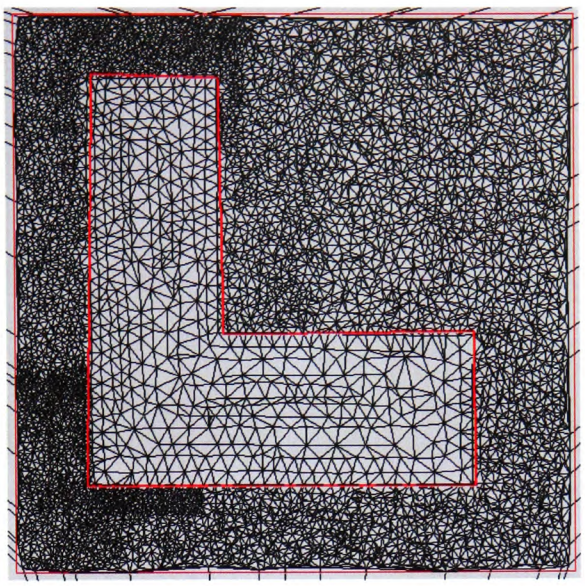


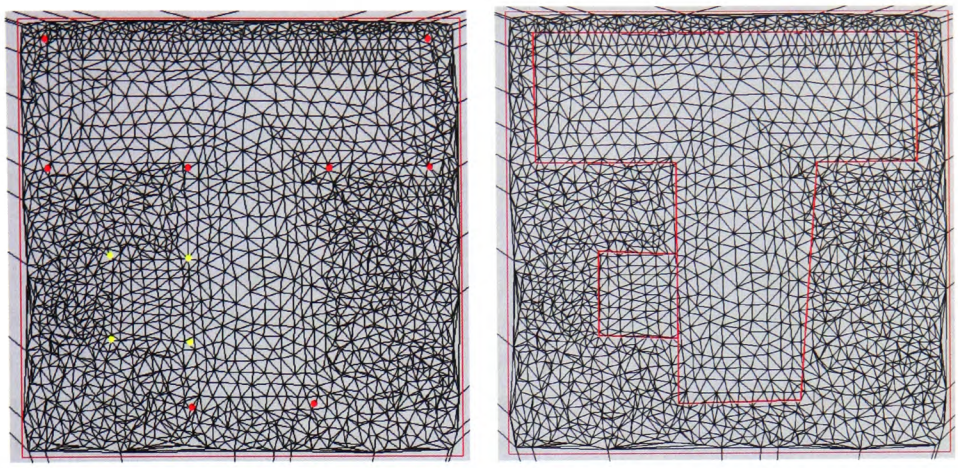
Figure 4.25: Building boundaries (thick red lines) are formed by connecting the corner points

In the case of a building extension, one more step is needed to create two building boundaries. Two sets of results are found when tracing the wall segments. They are used to form two sets of building corners. For example, Fig. 4.26(a) shows all the building corners from the wall segments in Fig. 4.19(a) (red and yellow points). The red building corners in Fig. 4.26(a) are found by analysing the red boundaries in Fig. 4.19(a). The yellow building corners, is the extension of the main building, are found by comparing the red and the blue boundaries in Fig. 4.26. The two set of boundaries are extracted to form two set of building corners. Points are deleted when they are too close (approximate less than 0.5 meter) to each other in the two set of corners; therefore all of the corners of the main building are removed. The remaining points are extracted to form the extension (yellow points in 4.26(a)). The next step is to use the red and yellow points to form the two buildings (Fig. 4.26(b)).

4.3.6 Procedures on Building Blocks Identification

All the programming steps in identifying building blocks using raw LIDAR data solely are:

1. Load raw LIDAR data points and create a Delaunay Triangulation
2. Sample the high density data points to a lower resolution.
3. Create a new Delaunay Triangulation using the lower resolution data points.
4. Check the big Voronoi cells which contain low and high points. Then calculate the eigenvector which has the smallest eigenvalue.



(a) The building corners of the main and extended buildings

(b) Building boundaries of the main and extended buildings

Figure 4.26: Searching a closed loop of the wall segments

5. Use the Statistical-Testing method between the high and low land and form a building wall segment (split Voronoi edge).
6. Connect the found wall segments to form closed building boundaries in clockwise and anti-clockwise directions.
7. Extract all the building wall segments and plot them on a circle.
8. Cluster the segments according to their orientation and geographical location.
9. Find a best fit line in each cluster of segments by using the orthogonal regression method.
10. Calculate the building corners by using a line to line intersection.
11. Compare the building corners and check whether there is a building extension.

4.3.7 Advantages and Disadvantages of the Statistical-Testing Method

The Statistical-Testing method shows the building block to be identified succesfully using the second pulse LIDAR data points only. Without using any other additional data source, it makes the system more independent. It is an advantages if there is only LIDAR avaliable for building reconstruction.

Preliminary building corners are found in this stage. Other set of building points will be found by intersecting roof planes and vertical walls in the next chapter. By comparing two sets of results, the accuracy of the building corner may be improved.

However the method is not able to find buildings which are built next to a hill or slope. The split Voronoi edges are not able to form a closed loop to represent a building; therefore it does

not work. It is possible to adjust the rule that a building is formed from an opened loop of wall segments but this has not been attempted in this work.

4.4 The Voronoi Mosaic City Model

The Voronoi Mosaic City Model is used to find the building blocks which try to separate the high building blocks (high land) from the surrounding terrain surface. It is based on the Voronoi Diagram which is the dual of the Delaunay Triangulation. Section 4.4.1 shows the advantages and disadvantages of using it. A Delaunay triangulation is created using the raw data points, and an index layer of the Voronoi diagram is created by sampling the data points into a lower resolution.

Raw data points (green in Fig. 4.27) inside each big Voronoi cell are extracted for comparsion. Dobashi et al. (2002) tried to move the Voronoi cells by comparing the colour value (Red, Green and Blue - RGB value) in a picture to form a mosaic image. They tried to move the Voronoi cells closer with similar RGB values and separate the cells with different RGB values. The same method is used to compare the elevation of the points. The Voronoi cells with similar heights will move closer to each other. Then the intervals between the Voronoi cells of the building (high z-value) and the Voronoi cells of the terrain (low z-value) will be maximised. The process will be repeated until a satisfactory end condition is achieved. In each time of the iteration, the limitation of the movement of a Voronoi cell will be fixed to its adjoining eight pixels.

A Voronoi cell is used to illustrate how to compare and move its cell boundary. The procedures include:

1. Take a Voronoi Cell and its centre point (sample red point in Fig. 4.28).
2. Move to its adjacent green points in four different directions and the Voronoi cell boundary will shift. Fig. 4.28 shows the old (red dash line) and new (green solid line) Voronoi boundaries and points out new points outside (blue) and inside (yellow) the boundaries.
3. Take all the points inside the Voronoi cell and calculate an averaged height value.
4. Compare the value to its adjacent Voronoi cell (sample point).
    - Move it back to its original location if it does not maximise the differences between its cell and its surrounding cells,
    - Stay in its moved location if it maximizes the differences

The moving and comparing processes are iterated until the calculated result among the Voronoi cells is within a tolerance. Fig. 4.29 shows the original Voronoi model and Fig. 4.30 shows the Voronoi model after three iterations. It roughly shows the building shape. The Voronoi City Model is not used in this project because the method is not accurate enough to produce building

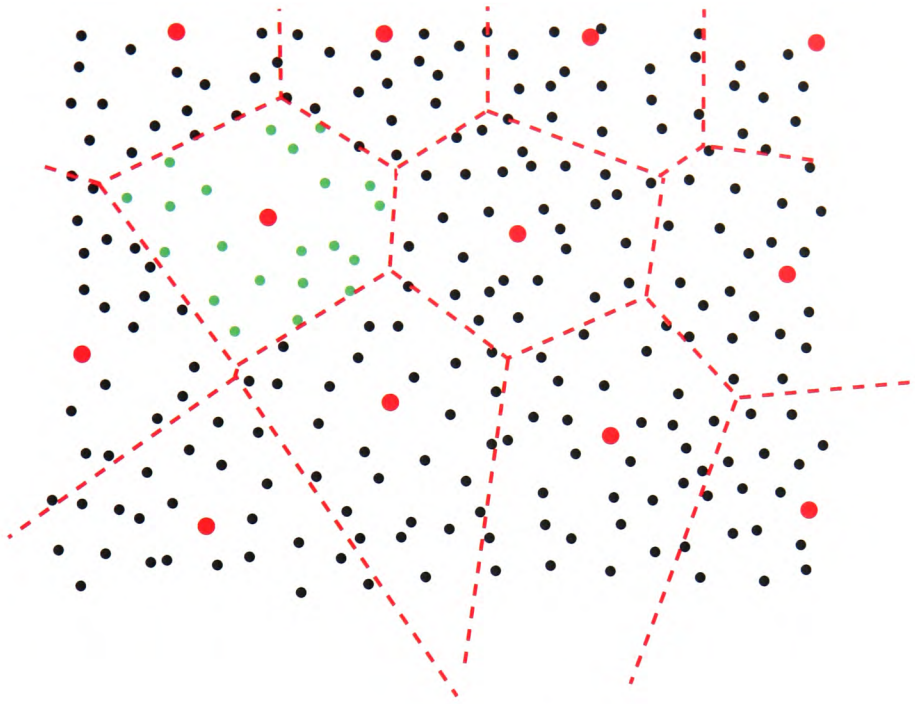


Figure 4.27: Raw data points (blue) are extracted for comparison

boundaries. However I believe that it could be useful if there is further development. The next section shows the advantages and disadvantages of this method.

4.4.1 The Advantages and Disadvantages of Using the Voronoi Mosaic City Model

The Voronoi Mosaic City Model may not be the best solution, but it definitely provides a new and different idea to solve the problem. One of the advantages of using this method is that there is no need to rely on additional data sources. This method allows the user to use raw LIDAR data only which is more flexible and independent. The second advantage is the connectivity between the buildings and the terrain surface. Ideally, after all the high Voronoi cells are found they will be merged to represent the building. Then Euler Operators are used to extrude the building from the terrain surface.

Two disadvantages of using the Voronoi Mosaic City Model were found. The first is that it is time consuming because the calculations between Voronoi cells are iterated. The iteration stops only when the result is within a certain tolerance. To improve the method, calculating of those Voronoi cells can be stopped when their results stop changing. The other disadvantage is that it not easy to get a straight building boundary from the Voronoi diagram. Fig. 4.30 shows

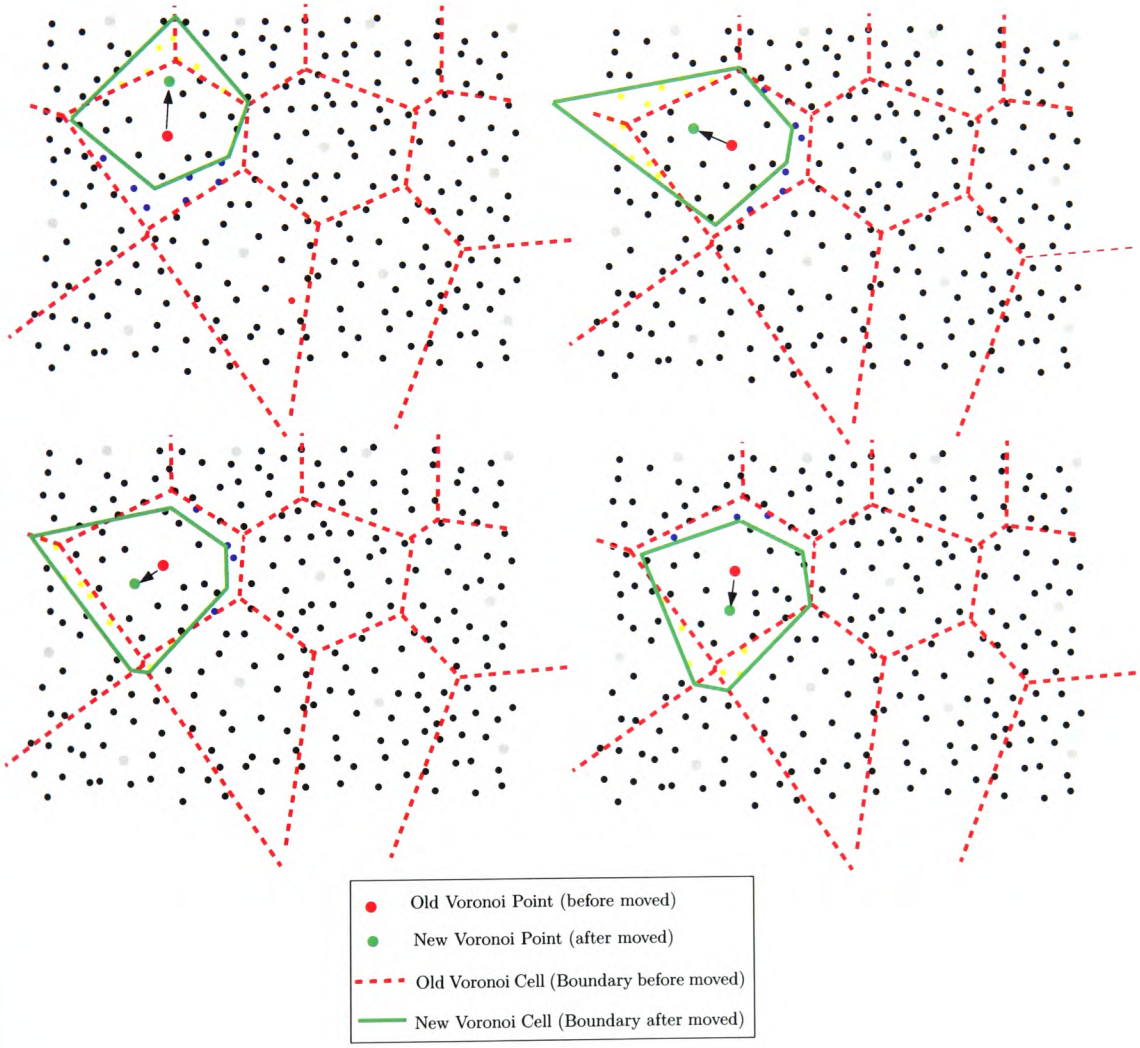


Figure 4.28: The red point can move in various directions (green point) which moves its cell boundaries (green solid line)

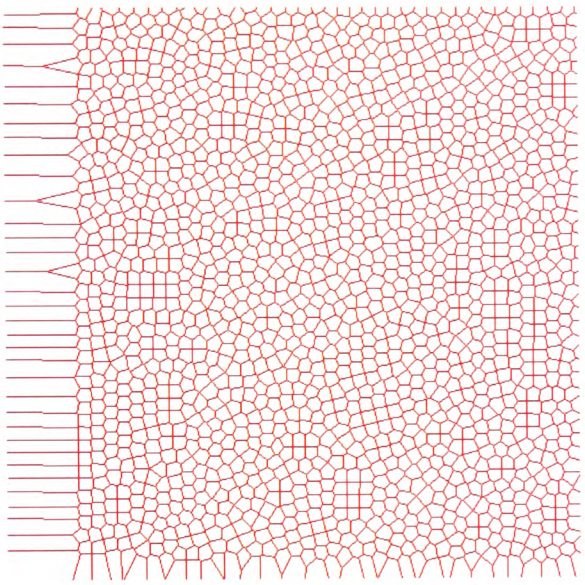


Figure 4.29: Original Voronoi cells

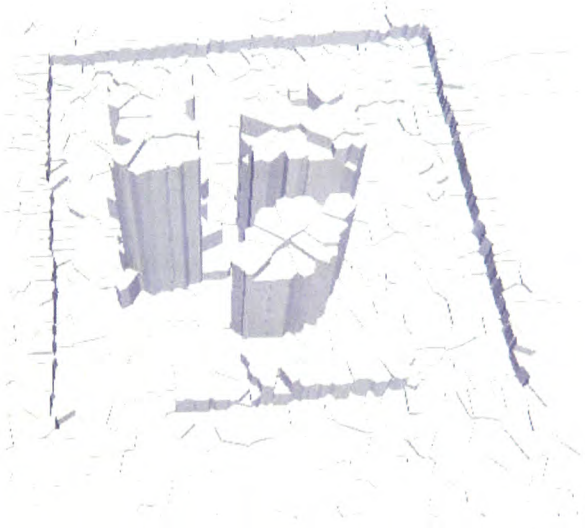


Figure 4.30: Voronoi cells after 3 iterations

the Voronoi model after three iterations, but the building boundaries are not sharp enough. A straight building boundary may be found by using an average Voronoi edges method. Further development is needed to improve this method for future use.

4.5 Ordnance Survey Mastermap Topographic Layer

Ordnance Survey (O.S.) Mastermap Topographic Layer is digital map data surveyed and shown by accurate surveyed positions of natural (e.g. vegetation) and man-made (e.g. buildings or roads) features on the terrain surface. These features are stored in different layers in a database. Building boundaries in one of the layers contains all the information about the buildings. Building corner points are extracted from the layer and inserted on a bare-earth model. The model provides surface data without any man-made objects or natural features and is created by using filtered LIDAR data points.

Section 4.5.1 describes the procedures for automated reconstruction of flat-roofed buildings using O.S. data. Section 4.5.2 shows how to extract data points within the building boundaries for further building reconstruction. The last section shows the advantages and disadvantages of using existing O.S. Mastermap or any other 2D map data.

4.5.1 Reconstructing Flat-Roofed Buildings Using Existing Landline Data

The University of Glamorgan campus was chosen to show the reconstruction procedures. Details of the data are:

- Filtered first pulse LIDAR data were captured by Infoterra in March, 2003 and interpolated onto a 1m * 1m grid using Inverse Distance Weighting. The filtered data forms a bare-earth model with 10m * 10m resolution. The accuracy of the provided LIDAR data is within +/- 15cm in height.
- A building layer is extracted from the O.S. data in ArcView Shapefile format.

Several steps are involved in reconstructing flat-roofed buildings using O.S. data. First a TIN is created with the filtered LIDAR data (the bare-earth model in Fig. 4.31). Then O.S. Mastermap Topograhic Layer is added to the terrain surface. Points are added in between of the starting and ending points to form a strainght line in a TIN model which we can a line-tracing procedure (described below). Third, a point-location algorithm is used to find out the LIDAR data points for each building and calculate the average heights (z-value) of each building. The last step is to extrude buildings using CAD-type Euler Operators. Though in this stage they are flat-topped buildings, remodelling of the roof structures will be shown in the next chapter.

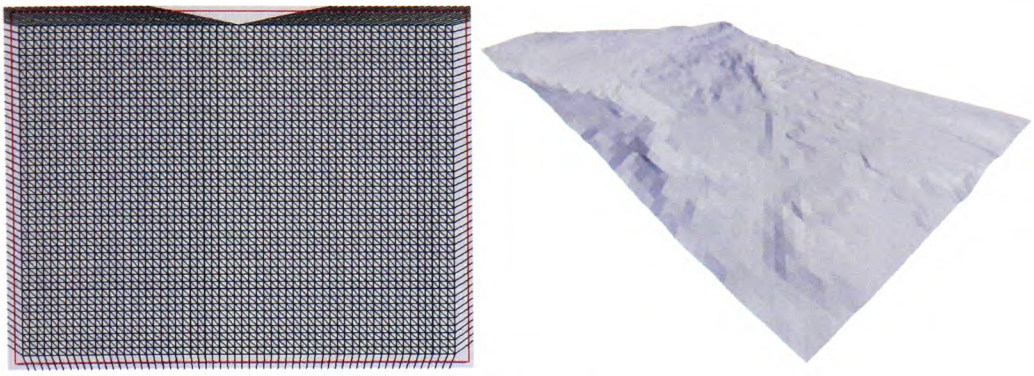


Figure 4.31: The 2D and 3D views of the bare-earth model

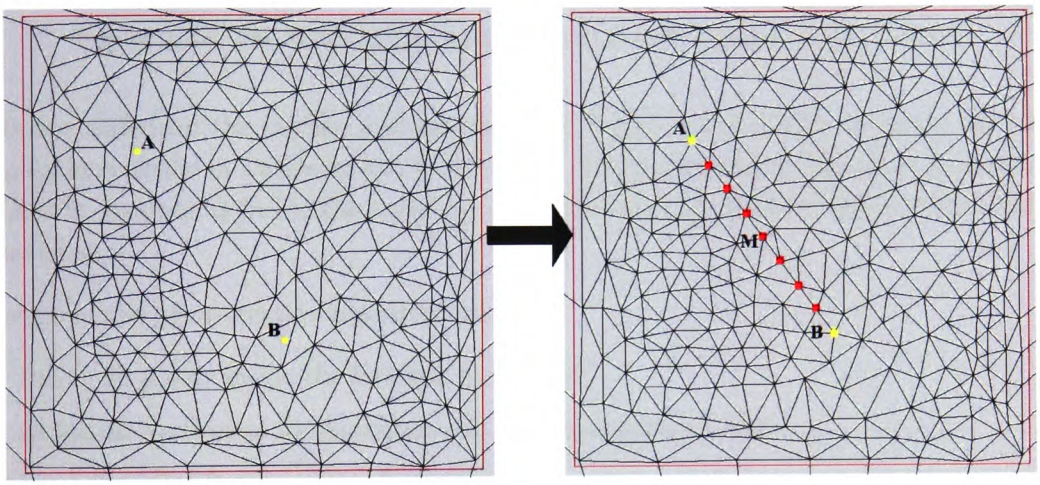


Figure 4.32: Connects points A and B by adding several (square shaped) points in between

A line-tracing procedure is used to insert building boundaries on the terrain model. There are several steps in the line-tracing procedure:

1. Insert two points on the terrain (Points A and B in Fig. 4.32)
2. Check if any previous triangle edge connects these two points
   - Stop if this is true, or
   - Insert a point half way between these two points if no edge connects them.
3. Repeat step 2 for each half of the line recursively until points A and B are connected by triangle edges.

The building footprint is not added on the terrain using the line tracing procedure in the next section because the building walls can be simplified. It is better not to use too many triangles to create the building vertical walls; therefore a constrained triangulation is used to add the building outline on the surface.

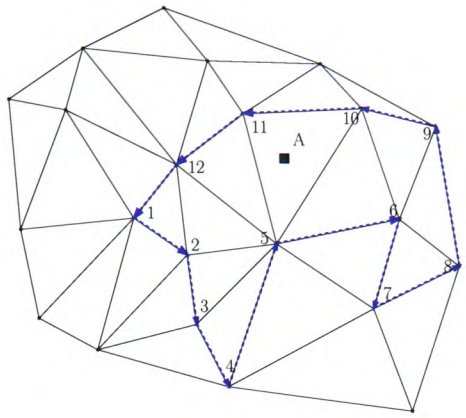


Figure 4.33: The order of the vertices for Point A is anticlockwise 5-10-11

Points are added on the surface and estimate the height of the points by using the surface interpolation method of Dakowicz and Gold (2002). The point-location algorithm is used to find the LIDAR data points inside the building boundaries and the average height of each building is calculated.

The steps of the point-location algorithm are:

1. Delete all points inside the building boundaries
2. Order the vertices of the boundaries in an anti-clockwise order
3. Load the LIDAR data points from the file
4. Walk through to locate the triangle which contains the LIDAR data point(s)
5. Check if the order of the vertices are anti-clockwise
   - Store the point in a list for further calculation if it is (point A in Fig. 4.33), or
   - Do nothing and go to another point if it is not anti-clockwise (point B in Fig. 4.34)
6. Calculate the average height in each list for the building height
7. Repeat from step 3 until all points are searched

This is a preliminary stage of automatic building reconstruction using existing Landline or cadastral data. It is a relatively easy building reconstruction method if all the data is available. However this may not be the case in the real world, especially the flat-topped buildings. The coming chapters and sections will show more complicated building reconstructions with different roof structures.

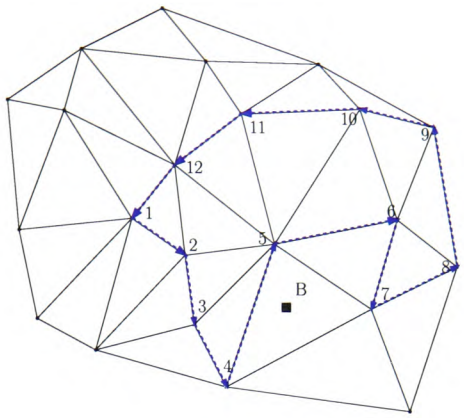


Figure 4.34: The order of the vertices for Point B is clockwise 4-5-7

4.5.2 Extracting Data Points from within Existing Building Boundaries

In this section all the LIDAR data points inside the building boundaries need to be extracted. The building boundaries have not been added on the terrain surface; therefore the point-location test cannot be used.

The first pulse LIDAR data points are used instead of a bare-earthed model because the points are required for finding the roof structure. First a terrain model is created using the first pulse LIDAR data. Then the O.S. data is loaded and displayed on the terrain model (the points are not added on the terrain model) in Fig. 4.35. The point-in-polygon test is used to extract data points which are inside the building boundary. The extracted building points are used to find out the roof structures. When the roof structures are found, all the building points will be deleted. The final step is to reconstruct the building from the TIN model with the roof structure.

The steps of the point-in-polygon test (Veblen, 1905) are:

- Find out the building bounding box (The maximum and minimum building corner points in Fig. 4.36)
- Take a point from the TIN model (points A and B in Fig. 4.37).
- Continue to check if the point is inside the building bounding box.
- Check the point whether is on the building boundary (line).

  yes The point is inside the building and end of the point-in-polygon test.

  no The point may be inside the building and continue the test.
- Draw a line (green line) from the point until it cuts the building boundary in Fig. 4.38.
- Count the number of intersection (red crosses) between the line and the building boundary in Fig. 4.38.

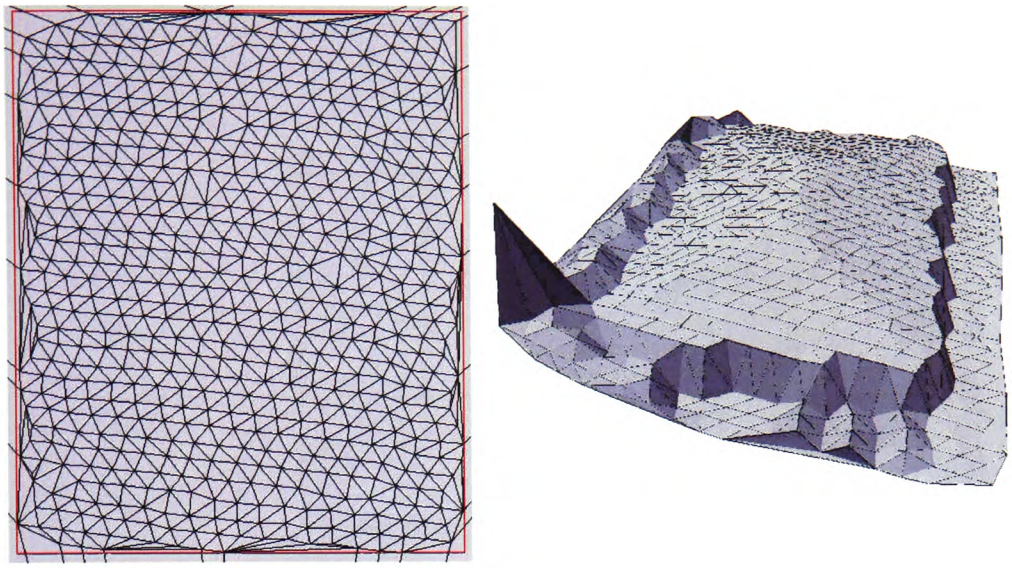


Figure 4.35: The 2D and 3D of a terrain model using First Pulse LIDAR data points

- Check the number of intersections
  - **a** The point is located inside the building if the number of intersections is odd (Point A in Fig. 4.38).
  - **b** The point is outside the building if it is an even number. (Point B in Fig. 4.38).

4.5.3 Procedures on Building Reconstruction Using 2D Map Data

All the programming steps of reconstructing a building with given ground plans are:

1. Load a bare-earth model, filtered LIDAR data, and use the data to create a Delaunay triangulation.
2. Load the given building boundaries and insert them on the terrain surface using the line-tracing algorithm.
3. Load the second pulse LIDAR data to calculate the building height for each building point-location algorithm.
4. Delete all the data points which are inside the building boundaries using the point-in-polygon test.
5. Extrude each building to its calculated building height.

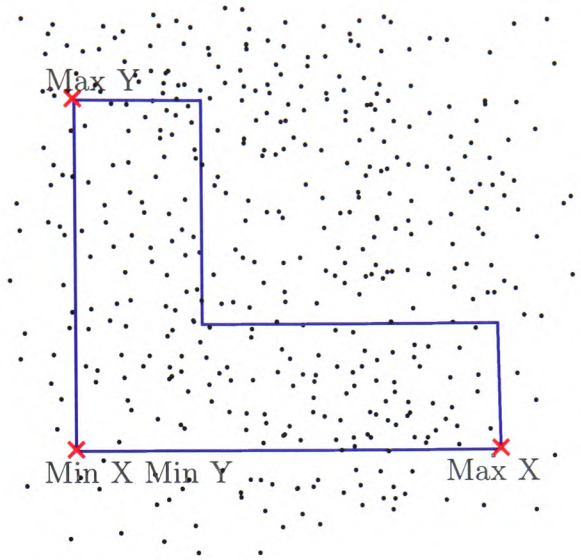


Figure 4.36: The building bounding box created by the maximum and minimum X and Y coordinates of a building (blue)

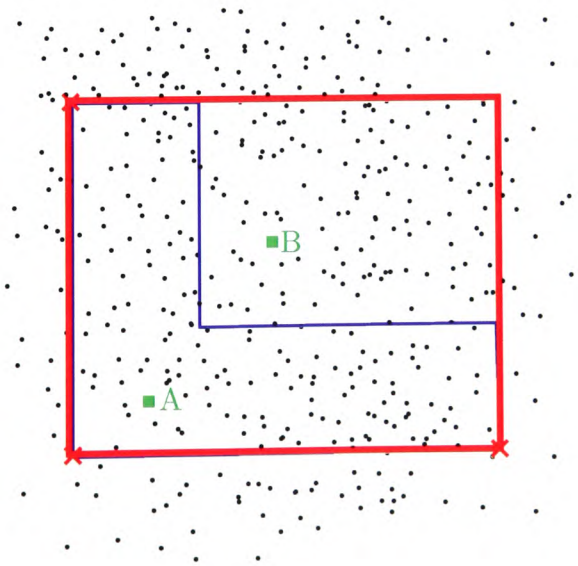


Figure 4.37: Points A and B are inside the building bounding box (red)

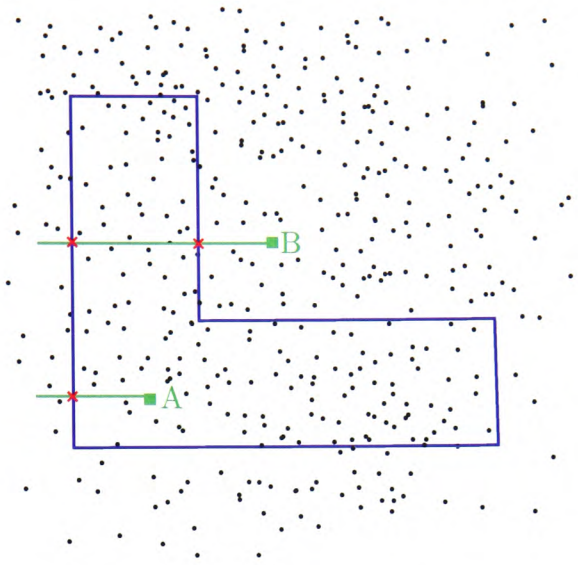


Figure 4.38: Green lines are drawn from the point to the building boundary and red crosses counted (A is inside, B is outside).

4.5.4 Advantages and Disadvantages of Using any 2D Map Data

Section 4.5.1 shows a simple idea on automatic building reconstruction using O.S. Mastermap Topographic Layer. If all data are available, and it produces a 3D overview map of a large area (like a small-scale map) with simple flat-roofed buildings, it may be an excellent solution. However it does not work if it needs to show roof structures.

Using existing O.S. Mastermap data to extract building points provides a reliable and simple solution. Though many researchers have tried to combine different data sources to extract building boundaries from LIDAR data, normally it seems too complicated or not accurate enough.

Many existing methods use additional data sources to find the building blocks. This has the disadvantage that the method may not be independent enough to solve the problem. A second disadvantage is that ground plans or 2D map data are captured based on the building footprints and not the corners of the roof. Normally the roof of the building is slightly or much bigger than the building footprint. A conflict or uncertainty occurs on whether or not to follow the building footprint (landline data) or slightly bigger roof (with unknown size) during the building reconstruction.

Sections 4.3 can be an alternative solution for finding the building outlines or separating the higher building blocks from the lower terrain surface. Raw LIDAR data are used to extract the building outline in the following method.

4.6 Chapter Summary

Building boundary extraction is the first step in building reconstruction. The location, orientation and shape of the building are identified in this procedure. The two methods developed to test the possibility of extracting the building boundaries are the Statistical-Testing method, and the Voronoi Mosaic City model.

The Statistical-Testing method is an alternative way to look for building boundaries, which separates the high and low areas by searching for the vertical wall segments. Then all the found segments are connected to form a closed boundary which is the building block. The segments are clustered according to their orientation and geographical location. Each group of wall segments are used to calculate an average vector. Building corners are formed by intersecting the averaged vectors.

LIDAR data is captured on an irregular pattern; this method works on a triangulation or a grid which increases the flexibility. No other pre-processing procedure is needed; therefore the originality of the data is preserved. This approach is data-driven; therefore presumably it works for different complex building shapes (e.g. hexagonal or even circular).

In this stage, the building corners found are in a preliminary stage and another set of building corners are calculated in the next chapter for comparison. This is an individual procedure to identify the building block. The next chapter describes the roof modelling technique which calculates the building corners again by intersecting roof planes and vertical walls. Comparison between two set of building corners can improve the quality of the results.

Though this technique works on a stand alone building only it is not be able to detect buildings next to a slope or hill at present, the problem can be solved by further development. It is definitely an effective and simple solution without using other external data sources.

Synthetic data, an L-shaped building and a T-shaped building with an extension on one side, are used to demonstrate and test the method. Chapter 7 shows some examples of using the last pulse LIDAR data.

The Voronoi Mosaic City model does not give a complete solution to the problem, but it shows the possibility of using a new method to separate the high and low areas. The method could be improved in the future.

In this chapter the building outlines or blocks are found, the next step is to look for roof types. All the points that fall inside the building will be extracted. Buildings have different roof types, for example flat, gabled or hipped. The further steps are required to find the roof structures. Chapter 5 shows two ways of roof modelling.

Chapter 5

Roof Modelling

Two technical terms are defined in the first section: normal vector, and normalised normal vector (unit vector). Then the introduction describes some current roof structures and gives a brief idea of the proposed roof remodelling method. Two different methods are shown in Sections 5.3 and 5.4. The final section gives a summary of this chapter.

5.1 Technical Background

Two technical terms are introduced which are normal vectors and normalised normal vectors. They are commonly used in this chapter. Principle Components Analysis (PCA) and MST methods are used in this chapter, they have been described previously.

5.1.1 Normal Vectors

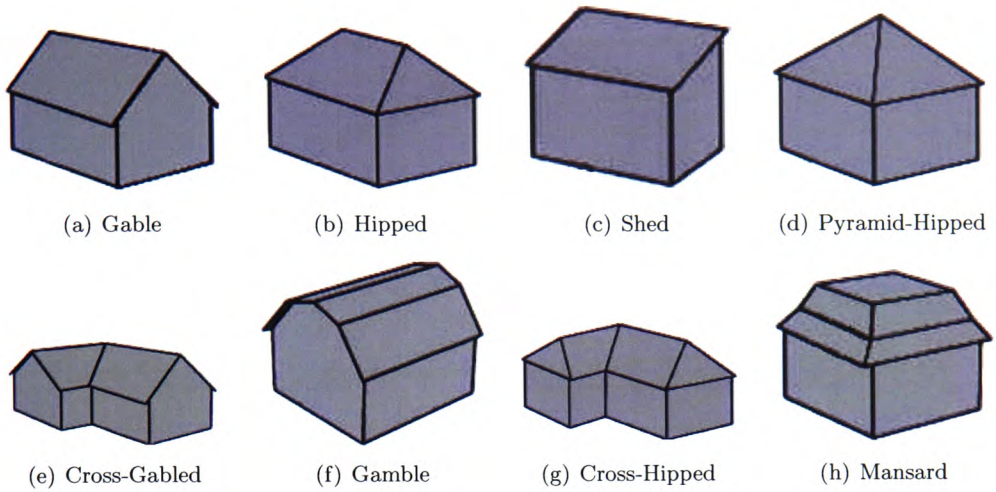
The normal vector, simply called the "normal" (Weisstein, 2007), to a surface is a vector perpendicular to it. A normal vector to a plane specified by

$$f(x, y, z) = ax + by + cz + d = 0 \tag{5.1}$$

is given by

$$N = \nabla f = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{5.2}$$

where ∇f denotes the gradient. The equation of a plane with normal vector $n = (a, b, c)$ passing through the point (x_0, y_0, z_0) is given by



(a) Gable (b) Hipped (c) Shed (d) Pyramid-Hipped

(e) Cross-Gabled (f) Gamble (g) Cross-Hipped (h) Mansard

Figure 5.1: Different Roof Types

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} = a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \tag{5.3}$$

5.1.2 Normalised Normal Vector(Unit Vector)

The "Normalised normal" vector, also called a unit vector, has a length of one (the unit length). A unit vector is often written with a superscribed caret or "hat", for example $\hat{N}$. The unit vector $\hat{N}$ having the same direction as a given (nonzero) vector N is defined by

$$\hat{N} \equiv \frac{N}{\|N\|} \tag{5.4}$$

where $\|N\|$ denotes the norm of N.

5.2 Introduction

Roof Modelling from raw LIDAR data is not an easy task because there are too many different types of roof, for example, simple gabled, hipped, cross-hipped and so on (Fig. 5.1). Some of the existing algorithms use a model-driven approach which extracts the roof information, and then searches for a best fit model from a library of building models (Brenner and Haala, 1998; Brunn and Weidner, 1997b; Haala and Brenner, 1999; Rottensteiner and Briese, 2003) to match the building footprint. However no pre-defined model is used in this project.

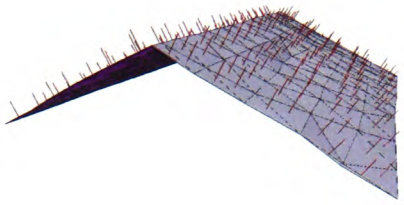


Figure 5.2: Interior triangles and their associated normal vectors(red lines)

A data-driven approach is used to extract the roof structure. The only assumption is that the roof is made up of planar segments. The information is extracted from the data points in order to form the planar faces. Two approaches are described in Sections 5.3 and 5.4. In both approaches, the interior points are extracted to model the roof structure, and then the extracted points are used to create a Delaunay Triangulation.

5.3 Simple Roof

Roof planes are extracted from the data points and intersected to find the ridges. A two plane simple gable roof is used to illustrate. This method works with simple roofs only; therefore another method will be introduced in the next section for more complicated roof structures. Artificial data is used to illustrate the idea in this section. The proposed approach is tested and evaluated using real data in Chapter 7.

5.3.1 Calculate the "Smallest" Eigenvector

Each of the interior triangles has an associated vector normal (the red lines in Fig. 5.2), which are perpendicular to the triangle surfaces. The gable roof looks like it has been formed by two forces squeezing from the left and right-hand sides. Therefore the left and right roof planes are folded. Charlesworth et al. (1975); Gold et al. (1981) used a PCA approach to find the fold axis which is used in this project. Normal vectors are used to calculate the "smallest" eigenvector using Principal Components Analysis (PCA), which is the same as the fold axis (in Fig. 5.3). The result is three eigenvectors with their associated eigenvalues. The eigenvector with the smallest eigenvalue is the "smallest" eigenvector.

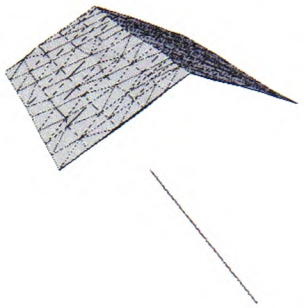


Figure 5.3: The "smallest" eigenvector

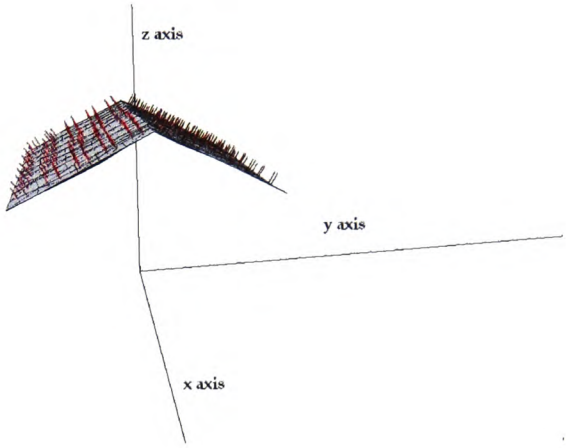


Figure 5.4: Normal vectors are projected on a two-dimensional coordinate system

5.3.2 Projected onto a Right-hand Coordinate System

The three-dimensional normal vectors are reduced to two-dimensions and projected on a two-dimensional coordinate system according to the "smallest" eigenvector in Fig. 5.4.

Normalise the normal vectors of the triangles to make all the vectors one unit length. Then they are plotted on the unit semi-circle (the red points are the normal vectors in Fig. 5.5), and most of the normal vectors fall between the values of 60 to 70 and 110 to 130. The plotted normal vectors are close to each other if they have the same orientation.

5.3.3 Cluster the Projected Normal Vectors (Triangles)

The normal vectors are then clustered into two groups according to their location (values) on the semi-circle. They are associated with the triangles; therefore two clusters of triangles are

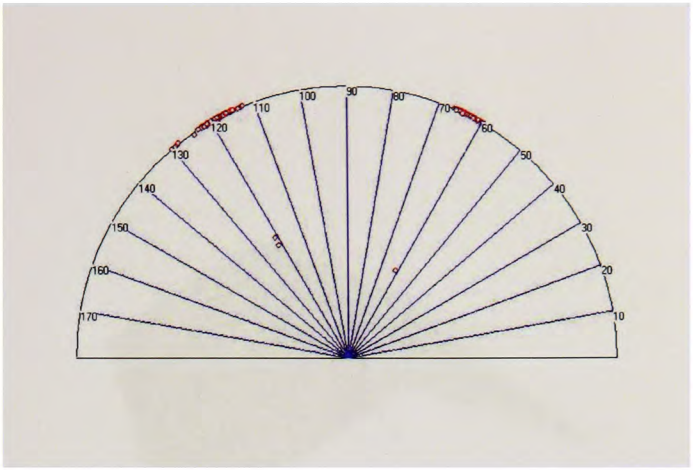


Figure 5.5: Normal vectors are projected on a right-hand coordinate system

found on the roof (red and yellow triangles in Fig. 5.6).

5.3.4 Roof Planes Creation

Each cluster of triangles represents a plane on the roof. Each group of triangles are extracted to find an average normal vector, Then two roof planes are created in Fig. 5.7. The two roof planes are shown for display reason and they are infinitie large. Plane intersection is used to extract the building corners and the ridge. The details of creating roof planes using the clustered triangles and intersecting roof planes are described in the following sections.

5.3.5 Procedures on Remodelling a Simple Roof

All the programming steps of extracting roof planes by plotting the normal vectors on a unit semi-circle are:

1. Extract data points inside the building boundaries.
2. Create a Delaunay Triangulation using the extracted points.
3. Take all the normal vectors (triangles) and calculate the eigenvectors, and eigenvalues using the PCA.
4. Extract the eigenvector which has the smallest eigenvalue.
5. Create a two-dimensional coordinate system using the "smallest" eigenvector.
6. Extract the triangles one by one and normalise their normal vectors.
7. Draw a semi-circle with lines to represent the angles.

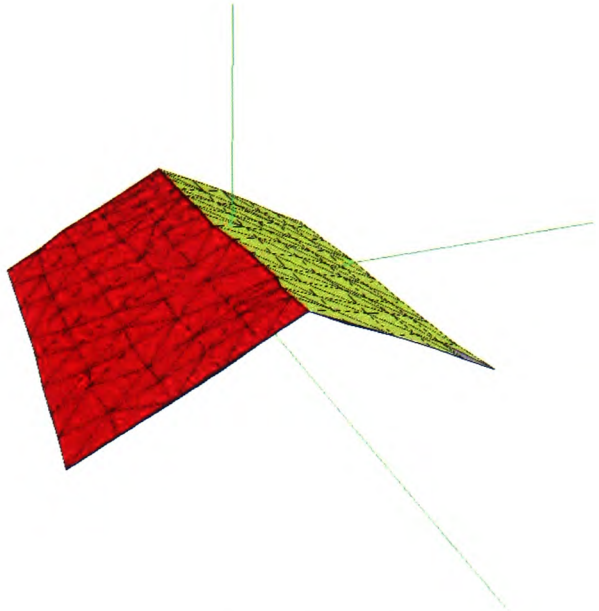


Figure 5.6: Clustered normal vectors with their associated triangles

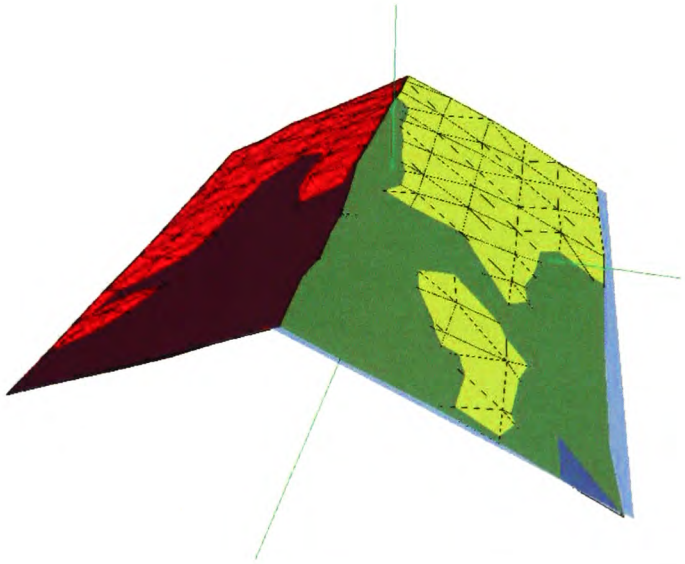


Figure 5.7: Two roof planes are formed using the clustered triangles

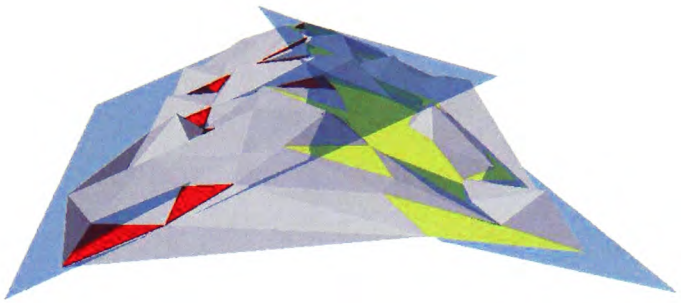


Figure 5.8: A clustered simple roof is created by noisy data

8. Reduce the dimension of the normal vectors and project them onto the coordinate system.
9. Plot the normal vectors on the semi-circle (the red points in Fig. 5.5).
10. Cluster the normal vectors according to the distance on the semi-circle.
11. Create roof planes using each cluster triangles (normal vectors).
12. Intersect the roof planes and the building boundary to create the building corners.

5.3.6 Evaluation and Conclusion on the Algorithm

Though the test data from the above example is synthetic data, the algorithm works well with real data even if the data is fairly noisy because the scattering of the normal vectors is fairly large. Fig. 5.8 shows the two clusters of triangles on the roof with noisy data. The advantage of using this algorithm is that perfect data is not needed.

The method seems to work well with noisy data, however it is not able to handle a roof with a complicated roof more than two planes. If there are two or more parallel planes on the roof, these may be separated at this stage by constructing the Delaunay Triangulation in x-y space for the data points of the cluster, extracting the MST, and separating the two or more parallel roof portions. Therefore the next approach is needed to solve the problem of extracting more complicated roofs.

5.4 Complicated Roof

This method works for roofs with more than two planes and complicated roofs. If a roof has many differently oriented segments, the normal vectors have to be projected onto the unit hemisphere. The approach works on a TIN structure and tries to extract roof planes by clustering

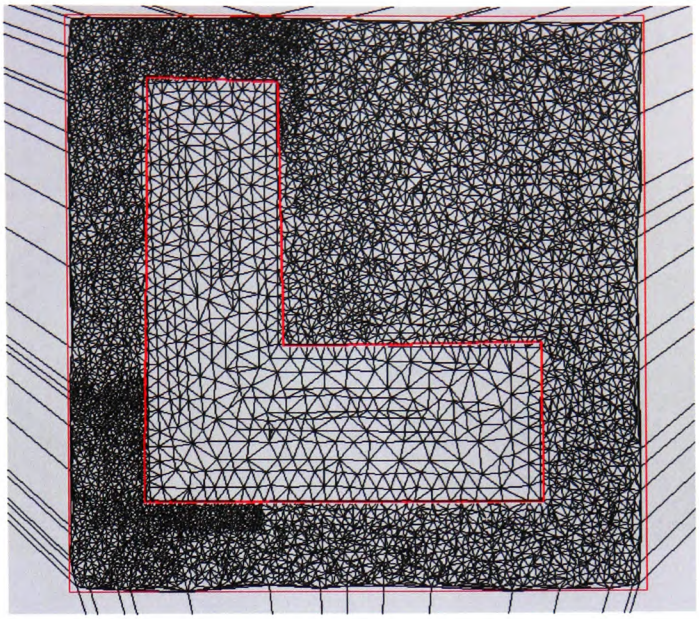


Figure 5.9: An L-shaped building example

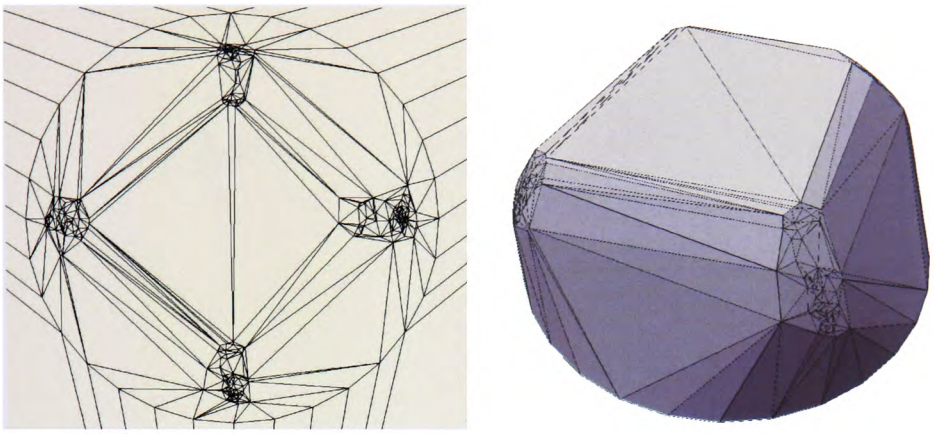
or extracting information from the triangles. A cross-hipped roof of an L-shaped building is used to describe the algorithm (in Fig. 5.9).

Three clustering methods are used to separate the triangles into groups. These are orientation clustering, perpendicular to orientation clustering, and geographical clustering. Region growing is used to absorb the missed triangles and search for the adjacency between clusters. Each group of triangles creates a roof plane. A three-plane intersection method is used to identify the roof and building corners. The algorithm that produces a set of corner points for building reconstruction is discussed in Chapter 6.

5.4.1 Orientation Clustering

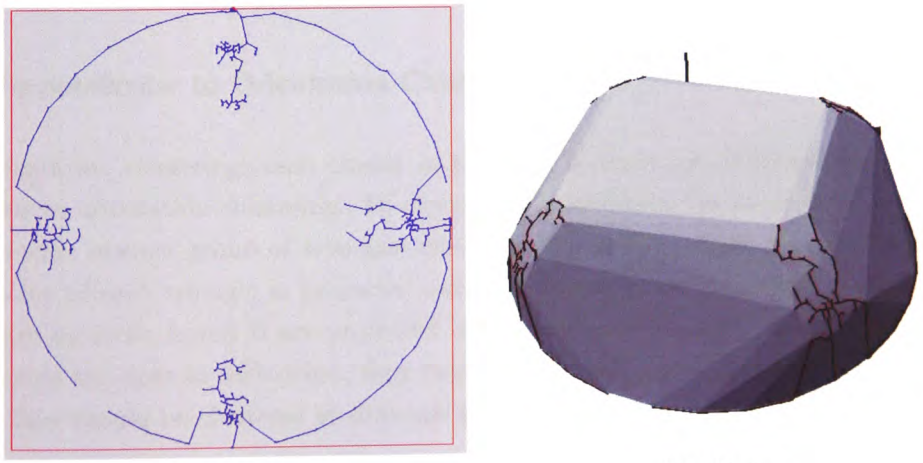
In Fig. 5.9 data points inside the red building boundary are extracted to create a Delaunay Triangulation. The next step is to plot all the normal vectors of the interior triangles on the unit hemisphere (2D and 3D views in Fig. 5.10). The plotted normal vectors show when the triangles have a similar orientation and if they are close enough.

The normal vectors on the hemisphere are connected using the MST in Fig. 5.11. The normal vectors that are clustered within a tolerance will be assigned to the same group. Groups having a small number of triangles are not going to be considered as a group in the system. Therefore in Fig. 5.12 only four big groups of clustered normal vectors are extracted for further processing.



(a) 2D view (b) 3D view

Figure 5.10: The plotted vector normal on the unit hemisphere



(a) 2D view (b) 3D view

Figure 5.11: The 2D and 3D views of the normal vectors connected using an MST

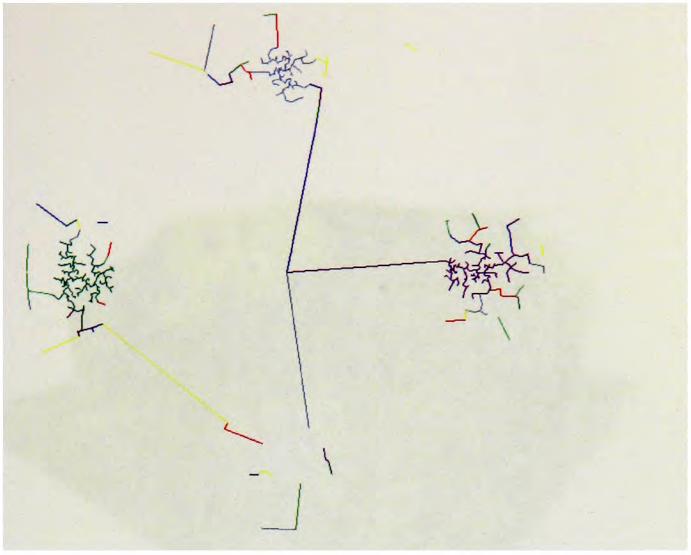


Figure 5.12: The clustered normal vectors on the unit hemisphere

Although normal vectors have the same orientation in the same cluster, they may not all be part of a single planar roof. Each cluster is then tested to see if it is part of a single plane, or of two parallel planes - if so the cluster is partitioned. Coplanar triangles or data points occur in separate roof portions: these must be partitioned again. Fig. 5.13 shows that the yellow and blue triangles on the roof have the same orientation, but they do not belong to the same roof plane. Further clustering is needed to group the triangles.

5.4.2 Perpendicular to Orientation Clustering

After orientation clustering, each cluster of triangles is extracted and grouped using the perpendicular to orientation clustering. To cluster triangles facing the same direction, an average normal vector of each group of triangles (the blue line in Fig. 5.14) is calculated. Then the centre point of each triangle is projected onto the averaged normal vector. The centre point of triangles on roofs A and B are projected onto the averaged vector normal. If the projected centre points are close to each other, they belong to the same roof plane and vice versa. If they are not, they should be clustered to different groups.

Normally the roof plane(s) of the building extensions have the same or similar orientation as the roof of the main building. The same method is used to separate the building extensions because there is a height difference between the extension and the main building. Fig. 5.15 shows that the triangles on roofs A (the main building) and B (the extension) have the same orientation. All of the triangles and theirs associated normal vectors are extracted to find an averaged normal vector. The averaged normal vector are reduced to a two-dimensional space,

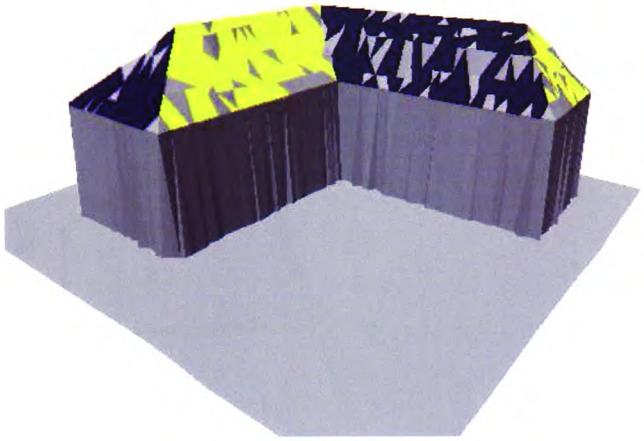


Figure 5.13: Coplanar triangles or data points occur in separate roof portions

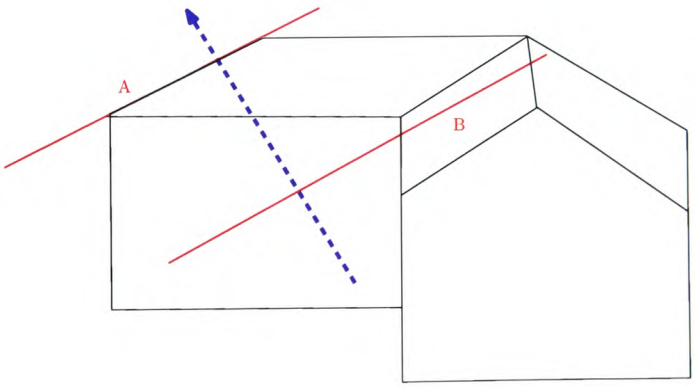


Figure 5.14: Triangles on roofs A and B projected onto the averaged normal vector (blue dashed line)

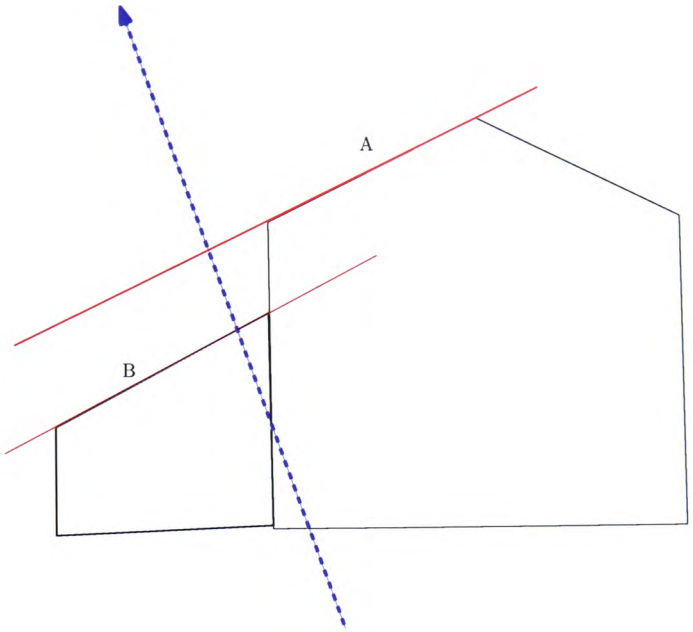


Figure 5.15: Triangles on roofs are projected on its averaged normal vector

and then the centre point of each triangle is projected onto the averaged normal vector (the blue dashed line). They can be clustered in two groups according to the projected locations which are far from each other.

5.4.3 Geographical Location Clustering

After the two clustering methods, some triangles can still be clustered in the wrong group. In Fig. 5.16 triangles on roof A and B are actually on two roof planes. They are still in the same cluster because they have the same orientation. Geographical location clustering is used to separate them. A Delaunay Triangulation is created using the centre point under the same group of triangles. The centre points are connected using MST and clustered according to their closeness. If the centre points of the triangles are close to each other, they will be clustered into the same group.

5.4.4 Region Growing

Region growing adds the unclassified triangles to the clusters to find the adjacency relationships between the triangle groups. 2D and 3D views in Fig. 5.17 show six groups of clustered triangles after the above three clustering methods. However some of the triangles are not classified in any of the groups. Region growing is used to regroup those unclassified triangles.

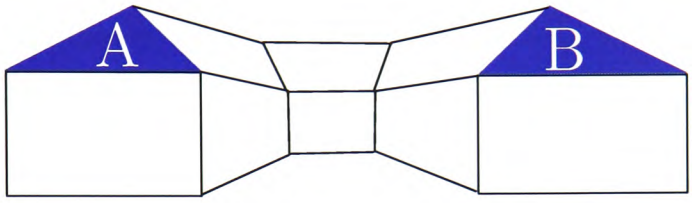
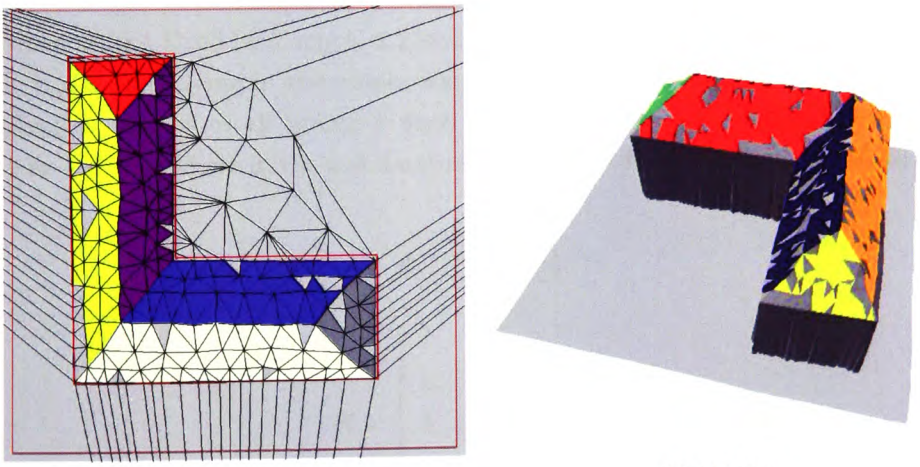


Figure 5.16: Triangles on roofs are projected on its averaged normal vector



(a) 2D view

(b) 3D view

Figure 5.17: The 2D and 3D views of the clustered triangles

Unclassified triangles are extracted and grouped according to the nearest classified triangle. If a triangle is in the middle of three triangles which are from three different groups, it classifies with one of the three groups. Though the ungrouped triangles will be classified to one of the groups, they will not be used for creating roof planes. It is not necessary that all triangles are classified, but the classification is mainly used for searching the adjacency relationship between triangle groups for three planes intersection as described in Section 5.4.6.

An example of a T-shaped building in Fig. 5.18 shows the region growing method. Fig. 5.18(a) shows the clustered triangles before using region growing. Some unclassified triangles in the building are shown in Fig. 5.18(b). After the growing method, the unclassified triangles are regrouped according to their nearest triangles (Fig. 5.18(c)).

An L-shaped building in Fig. 5.19 shows the adjacency relationships between six groups of triangles. Blue is next to yellow and gray. Red is next to white and purple. Grey is next to yellow, red, and white. The relationships between the groups are used for plane intersection in Section 5.4.6.

The clustering procedures are the same when there is an extension attached to the main building. Triangles are extracted from the two boundaries and classified using the above clustering methods. Fig. 5.20 shows the 2D and 3D view of the clustered triangles on the roof.

5.4.5 Roof Planes Creation

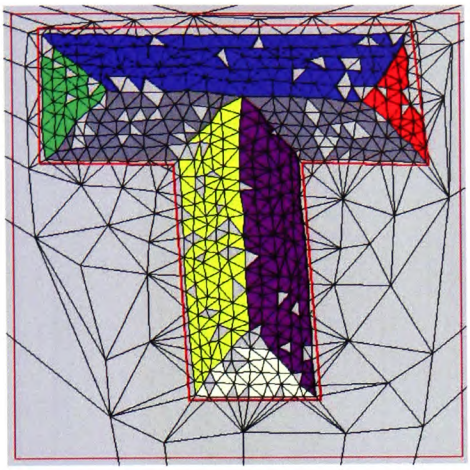
In three-dimensional Euclidean space, a plane can be defined by a specific point and a normal vector. P is the point lying in the plane, and $\vec{n}$ is a nonzero normal vector to the plane. The desired plane is the set of all points $\boldsymbol{r}$ such that, the plane is determined by the condition $ax + by + cz + d = 0$, where a, b, c and d are real numbers and a, b and c are not all zero.

$$\vec{n} \cdot (\vec{r} - \vec{p}) = 0 \tag{5.5}$$

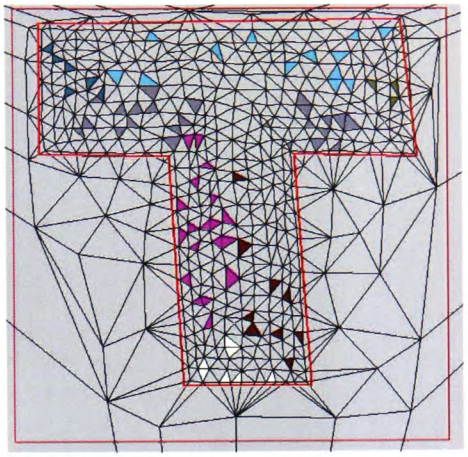
$$\vec{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, r = (x, y, z) \tag{5.6}$$

d as the dot product

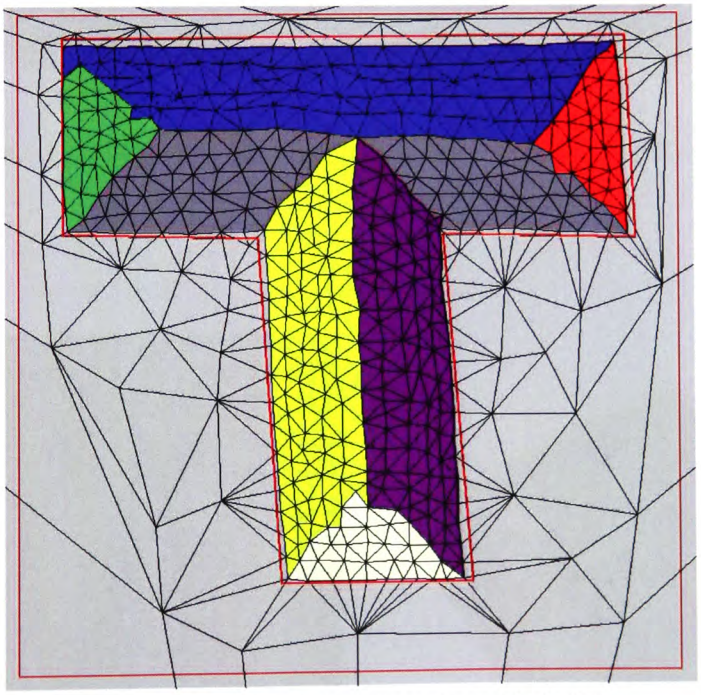
$$\vec{n} \cdot P = -d \tag{5.7}$$



(a) Clustered triangles of a T-shaped building



(b) Unclassified triangles in the building



(c) Seven groups of clustered triangles in the building

Figure 5.18: A T-shaped building is used to show region growing

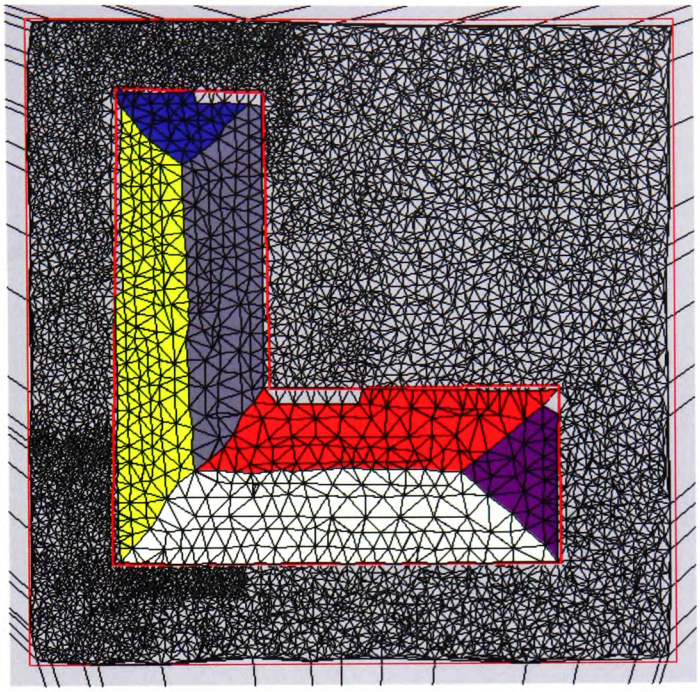
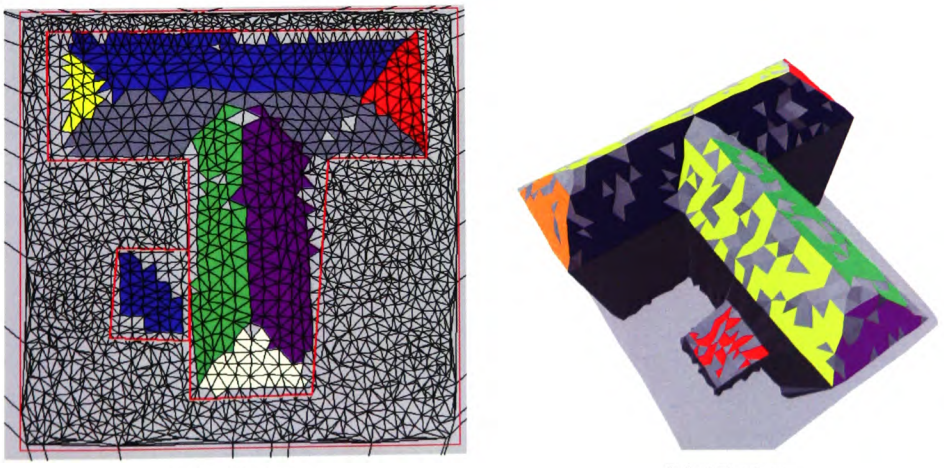
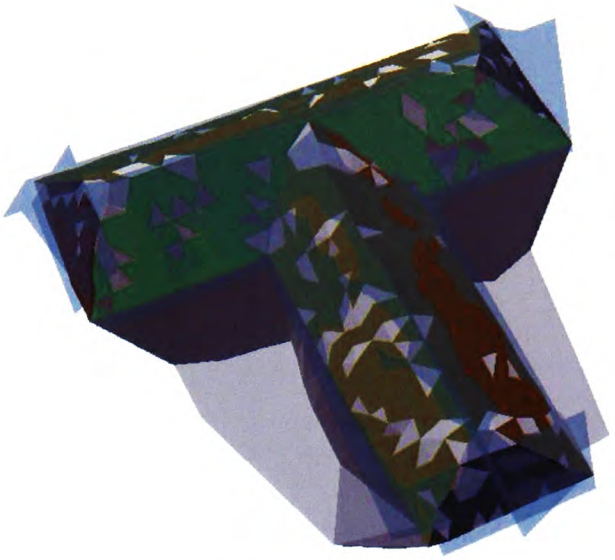


Figure 5.19: Six groups of clustered triangles of an L-shaped building

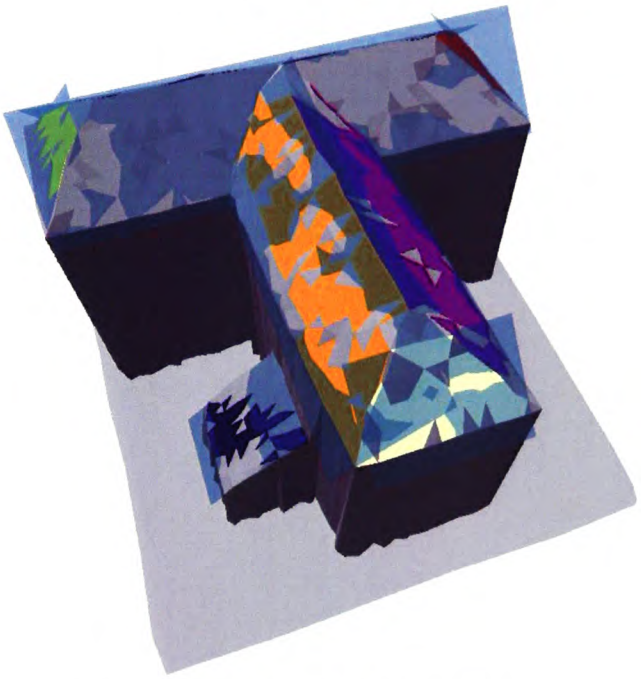


(a) 2D view (b) 3D view

Figure 5.20: Clustered triangles of a T-shaped and its extension buildings

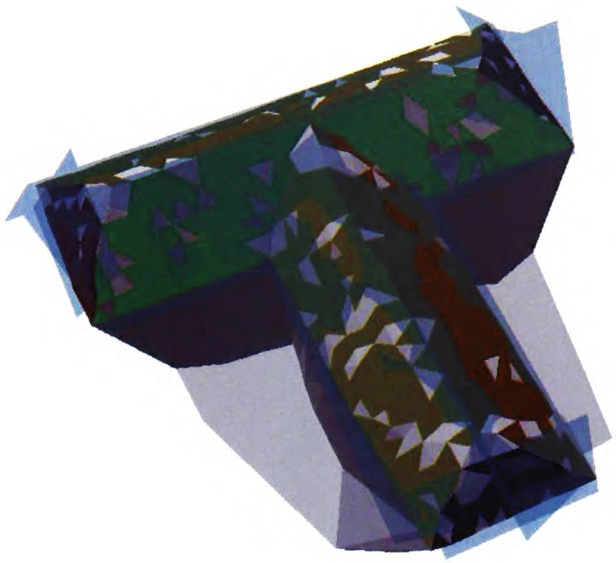


(a) A T-shaped building

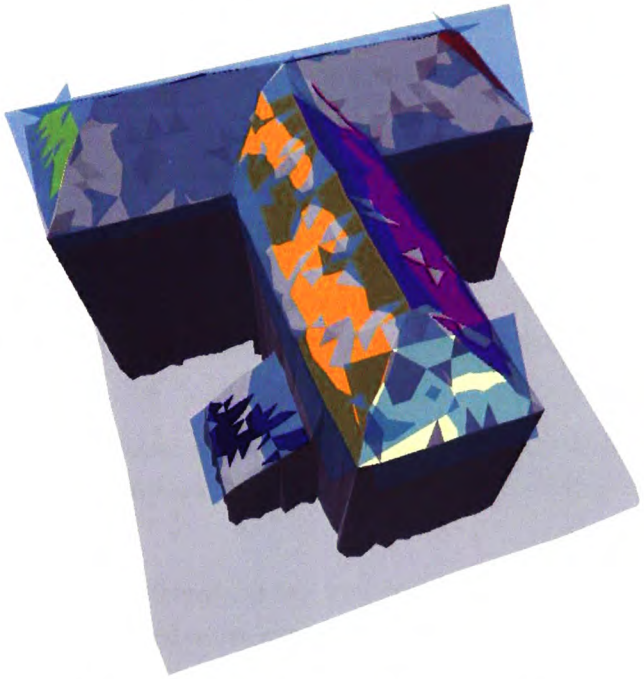


(b) A T-shaped building with its extension

Figure 5.22: Roof planes of two T-shaped buildings



(a) A T-shaped building

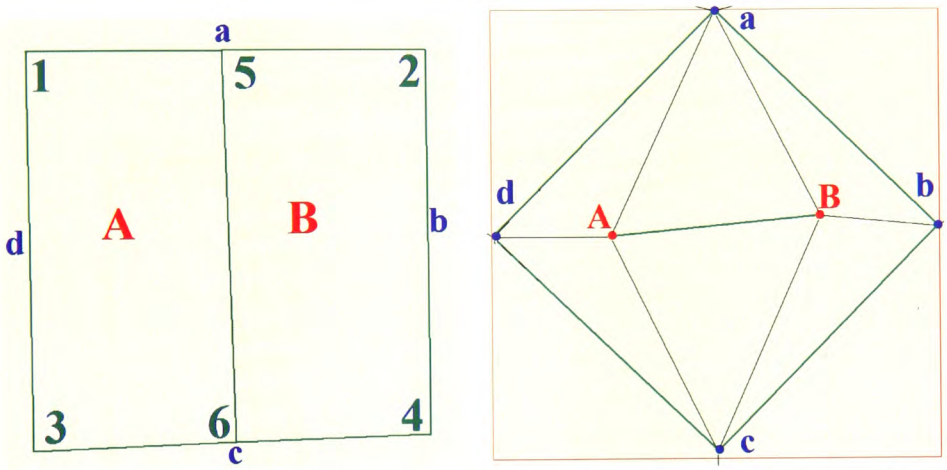


(b) A T-shaped building with its extension

Figure 5.22: Roof planes of two T-shaped buildings

Triangle (Fig. 5.23(b))	Intersected Points (Fig. 5.23(a))
ΔadA	Point "1"
ΔaAB	Point "5"
ΔaBb	Point "2"
ΔdcA	Point "3"
ΔAcB	Point "6"
ΔBcb	Point "4"

Table 5.1: Table of 2 planes gable roof



(a) Top view of the building

(b) Relationship between roof planes and building walls

Figure 5.23: Two planes gable roof

boundaries (an example of the 2 planes gable roof in Fig. 5.23). Some of the triangles from the simple DT may not be in a correct adjacency relationship. Region growing in Section 5.4.4 generates the adjacency relationships. Therefore the triangle edges are flipped until they match the relationships.

Four different types of roofs are used to describe the plane to plane relationship; these are two planes gabled roof, four planes hipped roof, hipped roof of an L-shaped building, and hipped roof of a T-shaped building.

A Two planes gabled roof is formed by two roof planes (A and B) and four building walls (a to d) in Fig. 5.23(a). Six triangles are created in Fig. 5.23(b). Point "a" in Fig. 5.23(b) represents the middle point of building wall edge "a" in Fig. 5.23(a). Point "A" (red) in Fig. 5.23(b) represents the middle point of roof plane "A" (red) in Fig. 5.23(a). Table 5.1 shows the relationships between the triangles and the building corners (Fig. 5.23). The Roof ridge is formed by the intersected points "5" and "6" in Fig. 5.23(a). Point "5" is the intersection between the building wall "a" and the roof planes "A" and "B" in Fig. 5.23(b) and Point "6" is the intersection between the building wall "c" and the roof planes "A" and "B".

Triangle (Fig. 5.24(b))	Intersected Points (Fig. 5.24(a))
ΔadB	Point "1"
ΔdAB	Point "1"
ΔACB	Point "5"
ΔBCb	Point "2"
ΔBba	Point "2"
ΔADC	Point "6"
ΔdcD	Point "4"
ΔdDA	Point "4"
ΔDcb	Point "3"
ΔDbC	Point "3"

Table 5.2: Table of 4 Planes hipped roof

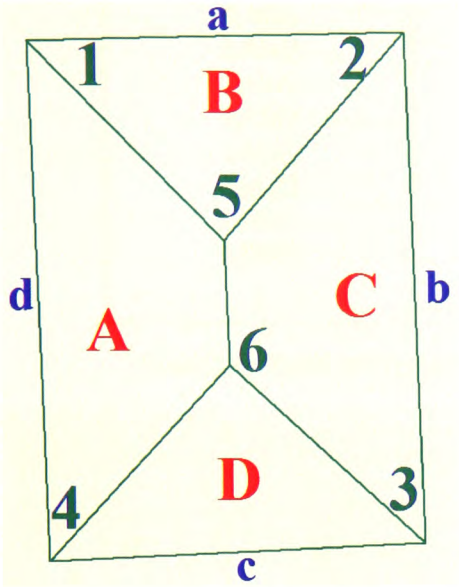
Triangle (Fig. 5.25(b))	Intersected Points (Fig. 5.25(a))
ΔafA	Point "1"
ΔfFA	Point "1"
ΔAFB	Point "7"
ΔABb	Point "2"
ΔaAb	Point "2"
ΔfEF	Point "6"
ΔfeE	Point "6"
ΔFEC	Point "8"
ΔECB	Point "8"
⋮	⋮

Table 5.3: Table of hipped roof of an L-shaped building

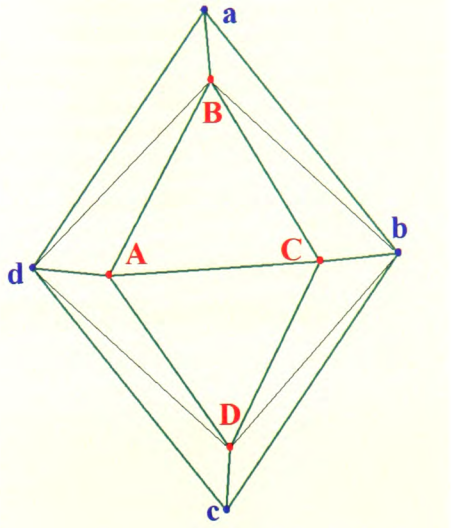
A Four planes hipped roof is created by four roof planes (A, B, C and D) and four building walls (1 to 4) in Fig. 5.24(a). Ten triangles are created in Fig. 5.24(b). Only six intersection points form the building because some of the planes intersect at the same point. For example ΔadB and ΔdAB, have the same intersection Point "1"in Table 5.2. Then Point "1" is averaged to form the building corner.

A Hipped roof of an L-shaped building is formed by six roof planes (A, B, C, D, E and F) and six building walls (a to f) of an L-shaped building in Fig. 5.25(a). Sixteen triangles in Fig. 5.25(b) create six building corners (Points 1 to 6 in Fig. 5.25(a)) and three ridge points (Points 7 to 9 in Fig. 5.25(a)). Table 5.3 shows the relationship between the triangles and intersected points.

A Hipped roof of a T-shaped building is formed by seven roof planes (A, B, C, D, E, F and G) and eight building walls (edges a to h) of a T-shaped building in Fig. 5.26(a). All the triangles in Fig. 5.26(b) creates seven building corners (Points 1 to 7 in Fig. 5.26(a)) and four ridge points (points 8 to 11 in Fig. 5.26(a)). Δagh and Δdec are created by three building vertical walls, two of which are coplanar; therefore they do not intersect at one point. The two

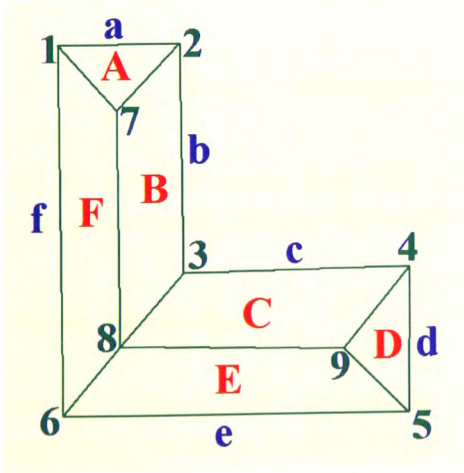


(a) Top view of the building

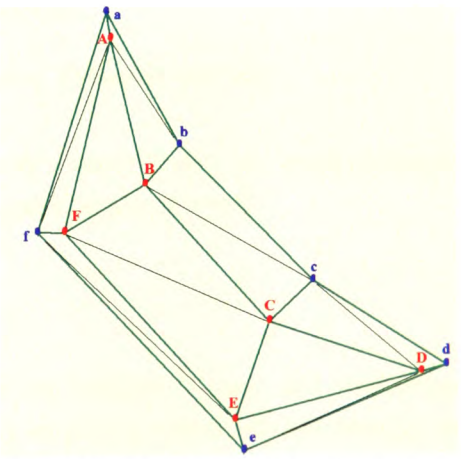


(b) Relationship between roof planes and building walls

Figure 5.24: Four planes hipped roof



(a) Top view of the building

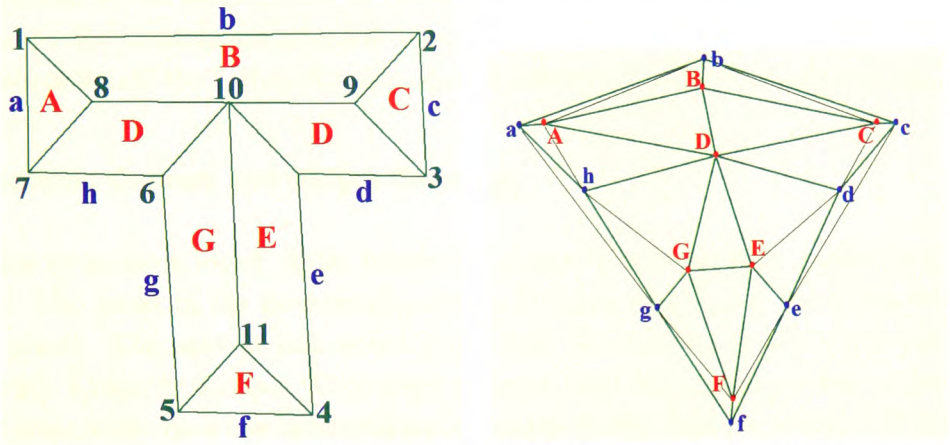


(b) Relationship between roof planes and building walls

Figure 5.25: Hipped roof of an L-shaped building

Triangle (Fig. 5.25(b))	Intersected Points (Fig. 5.25(a))
ΔaAb	Point "1"
ΔABb	Point "1"
ΔADB	Point "8"
ΔahA	Point "7"
ΔAhD	Point "7"
ΔhgG	Point "6"
ΔhGD	Point "6"
ΔBDC	Point "9"
ΔDGE	Point "10"
⋮	⋮

Table 5.4: Table of hipped roof of an T-shaped building



(a) Top view of the building

(b) Relationship between roof planes and building walls

Figure 5.26: Hipped roof of a T-shaped building

triangles are not used for searching for the intersection because they are invalid triangles. Table 5.4 shows the relationship between the triangles and intersected points.

Plane to Plane Intersection

Roof planes and building walls are used to find the building corners and roof ridges. Each valid triangle from the above section (without any co-planar planes) should have an associated three-planes intersection. Each intersection can produce a point, a line or no intersection. The relationships between the planes have been checked that the three planes are not coplanar. Therefore they will intersect at exactly one point.

The three planes can be written as:

$$N_1 \cdot P = d_1$$
$$N_2 \cdot P = d_2$$
$$N_3 \cdot P = d_3$$

where N_k is a unit normal for plane k, d_k is the displacement of the plane from the origin, and P is an arbitrary point.

The unique intersection point Q is given by:

$$Q = \frac{d_1(N_2 * N_3) + d_2(N_3 * N_1) + d_3(N_1 * N_2)}{N_1 \cdot (N_2 * N_3)} (Bourke,\ 2001) \tag{5.8}$$

The expression of the denominator is called the triple product of the three normals, and is equivalent to the determinant of a 3*3 matrix with those columns. The result of the three planes intersection of the L-shaped and T-shaped buildings are the blue points in Fig. 5.27.

5.4.7 Building Corners and Ridges Checking

The planes intersection found all the intersected points to form building corners and ridges of the roofs. The result of the intersected points is in pairs; therefore it shows the relationship between points. The pairs of intersected points form the building structure, for example the lines in Figs. 5.23(a) to 5.26(a). When there are more than three planes to form a corner point or a roof point, more than one intersection point can be found. Point A in Fig. 5.28 is averaged from the found intersection points.

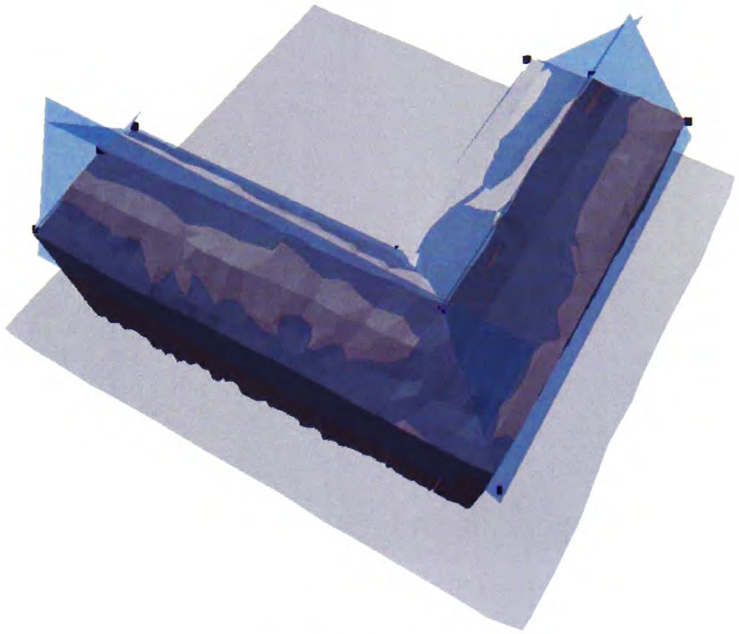
When there is an extension next to the main building, the extension points need to be added to the main building, for example Points B and C in Fig. 5.28. Building boundary MN in the main building is split into three pieces including Points B and C, which becomes MB, BC, and CN.

Fig. 5.29(a) shows all the building points including corner and ridge points. The roof planes are in a blue transparent colour. Another underneath view with all the intersected points are shown in Fig. 5.29(b) which includes some of the intersected points that cannot be shown in Fig. 5.29(a).

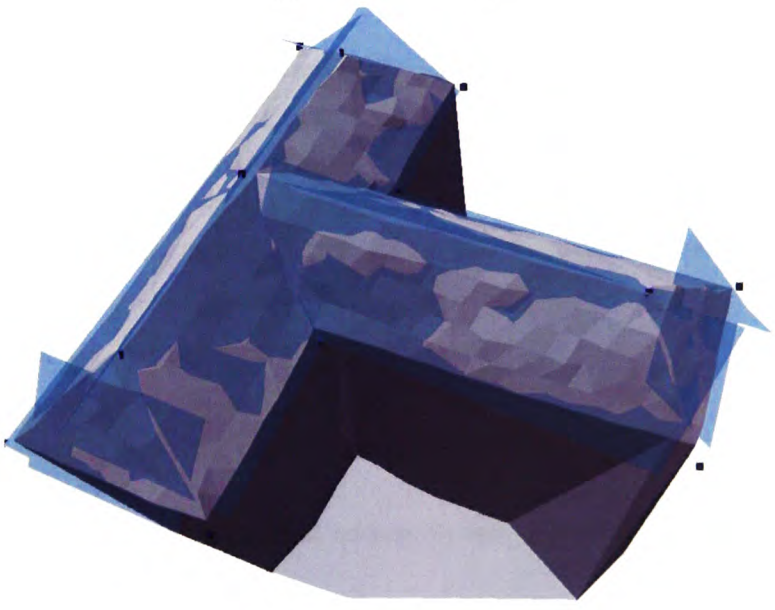
5.4.8 Procedures for Remodelling a Complicated Roof

The programming steps to extract roof planes using the three clustering steps are:

1. Extract data points inside the building boundaries.
2. Create a Delaunay Triangulation using the extracted points.



(a) L-shaped Building



(b) T-shaped Building

Figure 5.27: Result Intersection Points of the 3 Planes Intersection

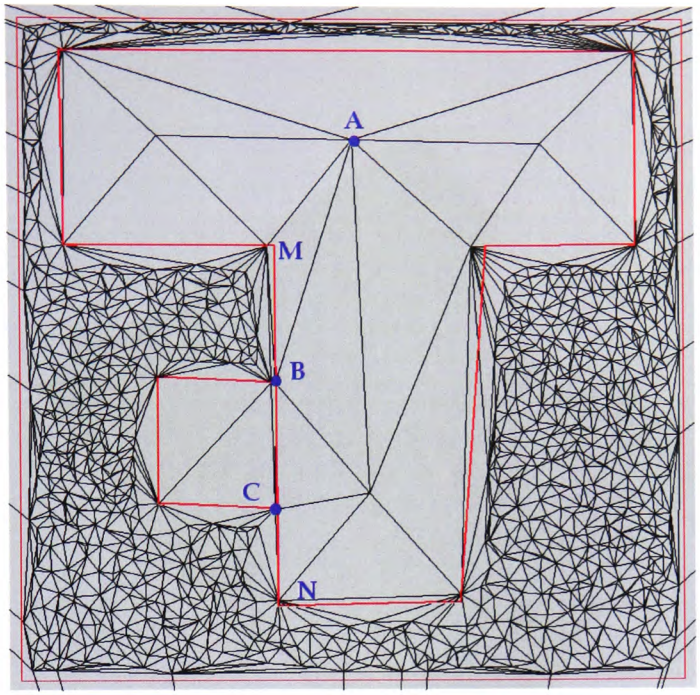
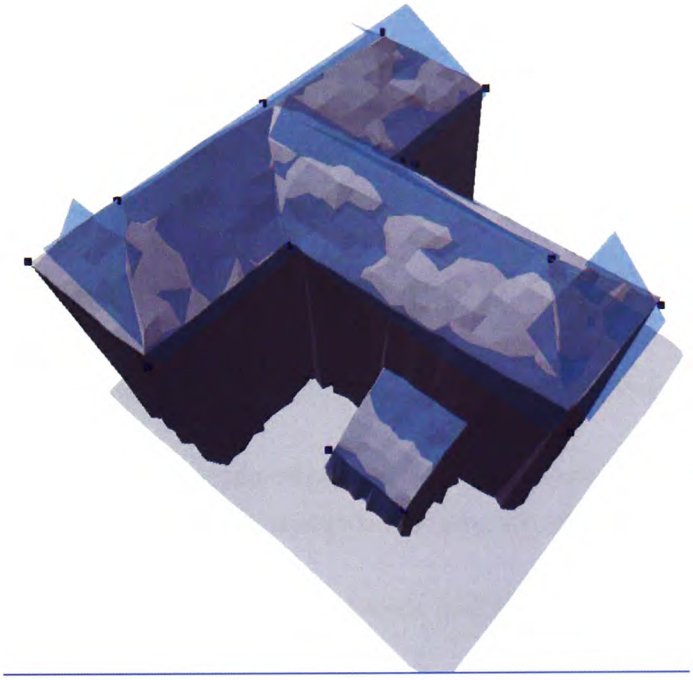
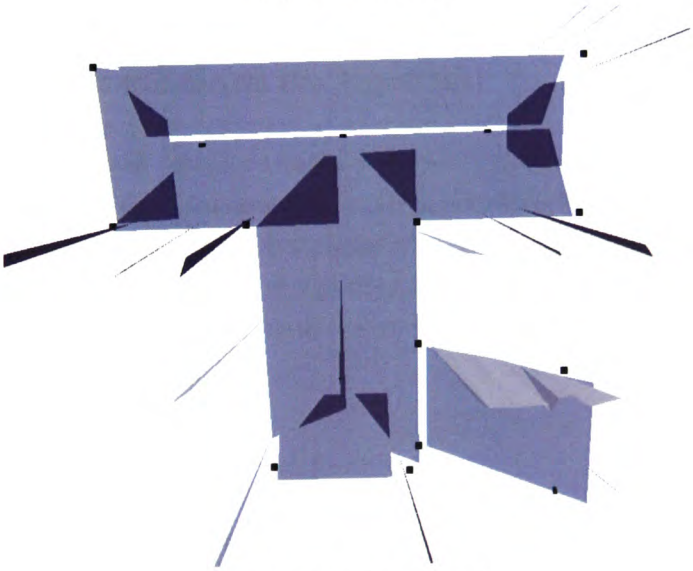


Figure 5.28: Intersected Points of the T-shaped building (with extension)

3. Perform Orientation Clustering
   - Extract the triangles one by one and normalise their normal vectors.
   - Create a unit hemisphere.
   - Plot the normal vectors on the hemisphere.
   - Connect the normal vectors using MST.
   - Cluster the normal vectors on the MST according to the distance between them.
4. Extract each cluster one by one.
5. Perform Perpendicular to Orientation Clustering.
   - Average the normal vectors of the triangles.
   - Extract the centre points of the triangles and project them onto the averaged normal vector one by one.
   - Separate the projected normal vectors in two groups if the normal vectors are far from each other.
6. Extract each cluster one by one.
7. Perform Geographic Clustering
   - Create a Delaunay Triangulation.
   - Extract the centre points of the triangles and insert them in the triangulation one by one.



(a) View from top



(b) View from underneath

Figure 5.29: Combining building corners and ridges of a T-shaped building

- Connect the inserted points using MST.
- Divide the inserted points in two groups if the normal vectors are far from each other.

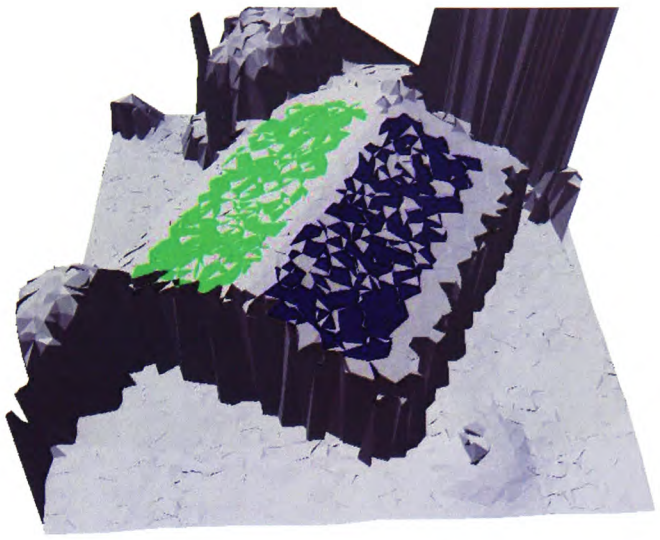
8. Perform Region Grouping to cluster the ungrouped triangles.
9. Search and save the relationship among the groups of clustered triangles.
10. Create roof planes on each group of clustered triangles.
    - Calculate an average normal vector for each group of triangles.
    - Find the centre, maximum and minimum points of each group of triangles for displaying the roof planes.
11. Intersect the roof planes and the building boundaries to create the building and roof corners.
    - Calculate the centre point of each group of triangles and the middle points of the building boundaries.
    - Create a simple DT using the calculated centre and middle points.
    - Check and flip the edges in the triangulation using the relationship between the groups of triangles.
    - Intersect the roof planes and vertical walls (building boundaries) according the triangulation.
12. Merge the intersected points when they are close to each other.

5.4.9 Evaluation and Conclusion on the Algorithm

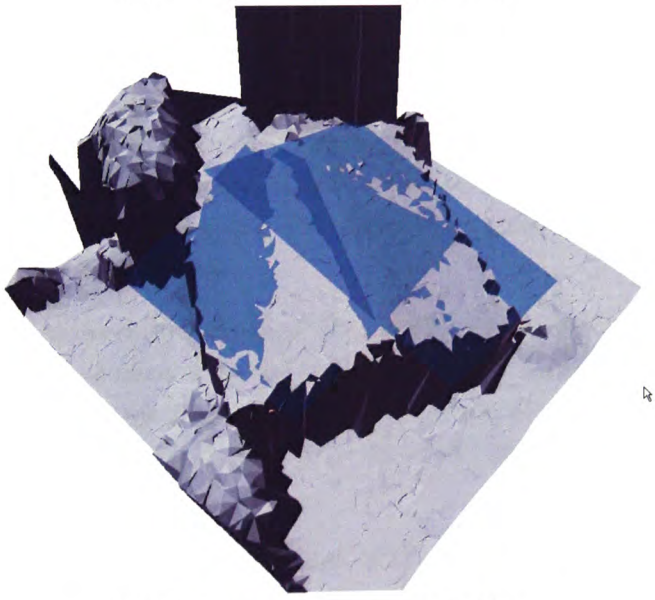
Fig. 5.30 is real LIDAR data which contains a simple gable roof structure. The clustering successfully searches the two roof planes on the building. However it works well with complicated roof structures. Real data of a roof with five planar surfaces is used to show the clustering result in Fig. 5.31. This method does not require a perfect data set to detect the roof planes. It allows more complicated roof structures (more than two roof planes).

5.5 Chapter Summary

Two roof remodelling methods are shown in this chapter. Both of them aim at detecting planar surfaces on the roof because I believe the roof structure is composed of roof planes. The simple roof method works on the gable roof structure only; however it is a simple and effective method. The second method works on complicated roof structures. Perfect data are not required when working on either method which is convenient and flexible for processing the data.



(a) Clustered triangles on a gable roof



(b) Roof planes on a gable roof building

Figure 5.30: An example of a gable roof building

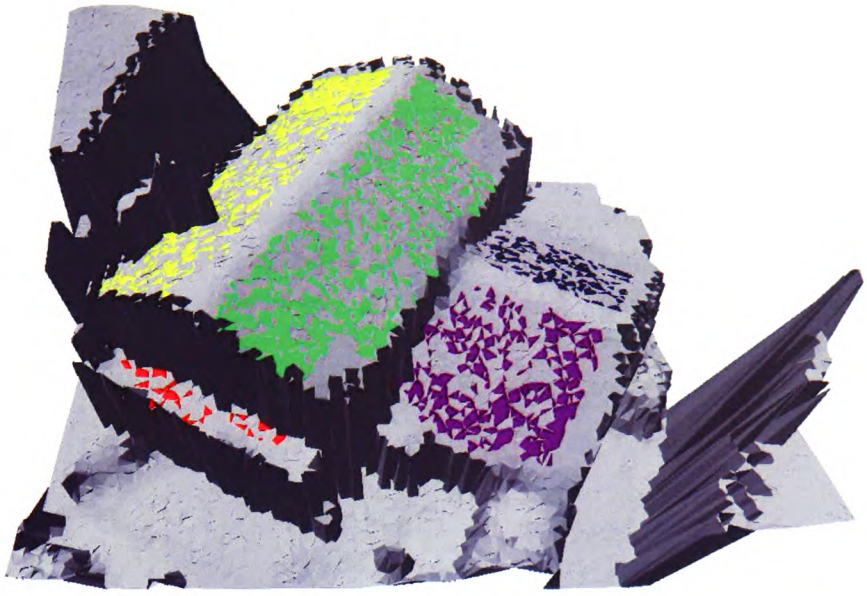


Figure 5.31: Real data of a roof with five planar surfaces

Both methods have similar approaches. They start by extracting data points inside the building boundaries to create a triangulation, and the triangles are clustered according to their properties. Each group of triangles represents a roof plane. Region growing is used to re-group the unclassified triangles, and then the adjacent relationships between roof planes and vertical walls are found. Building corners and roof ridges are calculated by using the three planes intersections between the roof planes and vertical walls. Though the two methods have similar approaches, their clustering algorithms are different.

In the first method, triangles inside building boundaries are extracted. Each triangle has a normal vector which is perpendicular to the triangle. The 3D normal vectors of the triangles are projected and plotted onto a 2D semi-circle. The projected normal vectors are clustered according to the distances between them. They are clustered in the same group if they are closed to each other.

Three clustering algorithms are used in the second method to separate the triangles according to orientation and geographical location. The first two clustering procedures separate the triangles according to their orientations. Step one looks for the orientation of the triangles. The second step checks whether the triangles with the same orientation are co-planar. The last one clusters the triangles according to their geographical location.

The second set of building corners are found by intersecting roof planes and vertical walls, and then they are used to compare to the building corners from the last chapter. The two sets of

building corners are averaged to improve the quality of the result.

Artificial data of an L-shaped and a T-shaped building with an extension at a side are used to test and show the methodology, however the coming chapter will show some examples of the method using real LIDAR data.

Although simple roof clustering works on simple roofs only, it gives reliable results on some noisy data sets. Complex roof clustering allows more flexibility for modelling different kinds of roof structures. Modern architecture creates more and more complicated roof structures. Following the research of Charlesworth et al. (1975), a complicated roof structure like the Wales Millennium Centre with an arch shaped roof may be able to be modelled in the future.

Chapter 6

Building Reconstruction and Program Manual

The first section gives the technical background which describes a term used in this chapter. Procedures for building reconstruction are described in Section 6.3. A Program Manual is shown in Section 6.4. The last section gives a chapter summary. An L-shaped and a T-shaped buildings are used to demonstrate the procedures.

The system has been developed using an object-oriented programming language - Delphi version 6.0. The user manual of the system is shown in Section 6.4. It shows fundamental steps for loading and displaying the terrain surface in 2D and 3D views, building reconstruction, further terrain or building modification.

6.1 Technical Background

A technical term, "Constrained Delaunay Triangulation", used to reconstruct the buildings, is shown in the technical background section.

6.1.1 Constrained Delaunay Triangulation (CDT)

The fundamental properties of a Delaunay Triangulation (DT) was explained in Chapter 4. Constrained Delaunay Triangulation (CDT) is the triangulation of the vertices with the following additional properties (Chew, 1987):

- certain found specific edges to be included in the triangulation.
- as close as possible to the Delaunay Triangulation.

A CDT is different from a DT because it does not always assume the Delaunay criterion, for example some of the triangles form circles which may contain other points inside. Therefore the empty circle criteria may not be true in CDT.

6.2 Introduction

Building Reconstruction extends a current two and half dimension to a three dimensional terrain model. Computer Aided Design (CAD) on the other hand is often fully 3D, which may be excessive for extended 2D mapping needs. Popular methods create and display the terrain using triangulation and the 3D buildings are put on top of the existing triangulation. This looks like playing "LEGO^{TM}". Since the DTM and the buildings are not in the same layer, it may be difficult or impossible to model tunnels or bridges on the surface with connected topology (Tse et al., 2004). Therefore building reconstruction in this project is based on the method in Section 3.3 which aims at preserving the topological connectivity. Then the extended TIN model can be easily modified in the future. L-shaped and T-shaped buildings are used to illustrate the procedures used to reconstruct the buildings.

6.3 Procedures for Building Reconstruction

When the building boundaries are located and the roof structure is found, the next step is to reconstruct the building. This method extrudes the building from the terrain surface: therefore all points inside the building must be removed. Then the building outline is added to terrain surface. Finally, Euler Operators are used to extrude and reconstruct the building.

6.3.1 Deleting Points Inside the Building Boundary

To remove all data points inside the building boundary, a point-in-polygon algorithm (as described in Chapter 4) is used. However, sometimes there may be an offset between the building outline found and the building itself. To avoid this happening the size of the building boundary will be slightly enlarged by about 0.5 m in each direction. The algorithm checks the data point by point: if the point is inside the enlarged boundary, it will be deleted. If the point is outside the building, it will be kept. The point deletion algorithm (Devillers, 1999) is used to remove points from the Delaunay Triangulation.

Fig. 6.1 shows the 2D view of the terrain model when the data points inside the building are deleted. The procedures for deleting points inside or near the building boundaries are:

1. Take the calculated building boundary.
2. Enlarge the building boundary by 0.5 m in all directions.
3. Extract all data points within the enlarged building bounding box and store them in a list.
4. Take a point at a time and check whether the point is inside the enlarged building bounding box. (Repeat until all the data points are checked).

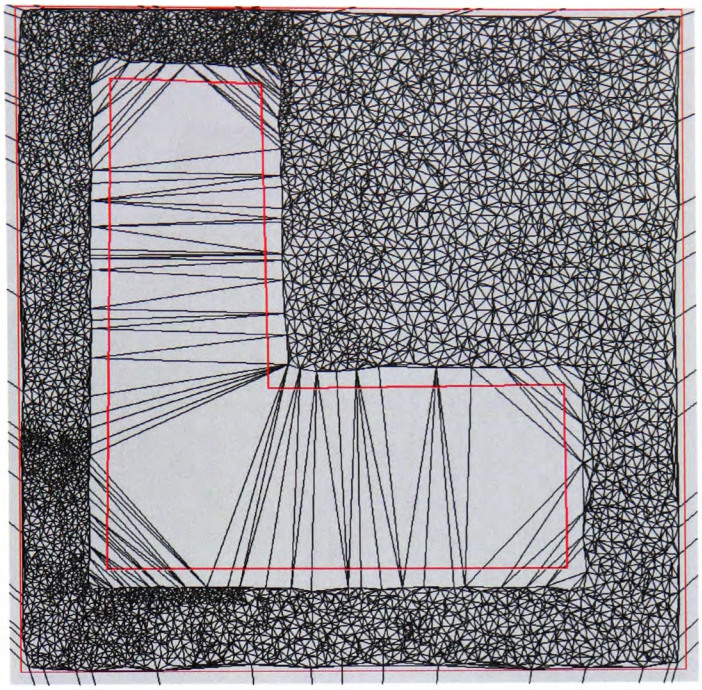


Figure 6.1: 2D View of terrain model after deleting the bulding points

5. If the point is inside the bounding box, check whether is inside the enlarged building boundary using the Point-in-Polygon algorithm.
6. If the point is inside the enlarged building boundary, delete it and if not retain it.

6.3.2 Insert Building Outline and Ridges

Building corners and ridge points are found using the three planes intersection method. Nine intersection points are found in the sample L-shaped building which are the blue points in Fig. 6.2. Six of them are the building corners and three are the ridge points.

The z-value of the intersected points is the building height, but the points must be added on the surface before the height can be extruded. An algorithm for estimating the height of the points from the terrain is used to find the z-value of inserting points on the terrain. Since the x and y coordinates of the intersection points are known; the surface z-value has to be found. Points are added on the terrain and estimate the height of the points by using the surface interpolation method of Dakowicz and Gold (2002).

The points are added on the terrain surface using the Constrained Delaunay Triangulation. CDT is used because it simplifies the form of the building. If the points are added using the line-tracing algorithm, too many triangles are used to form the sides of the vertical building

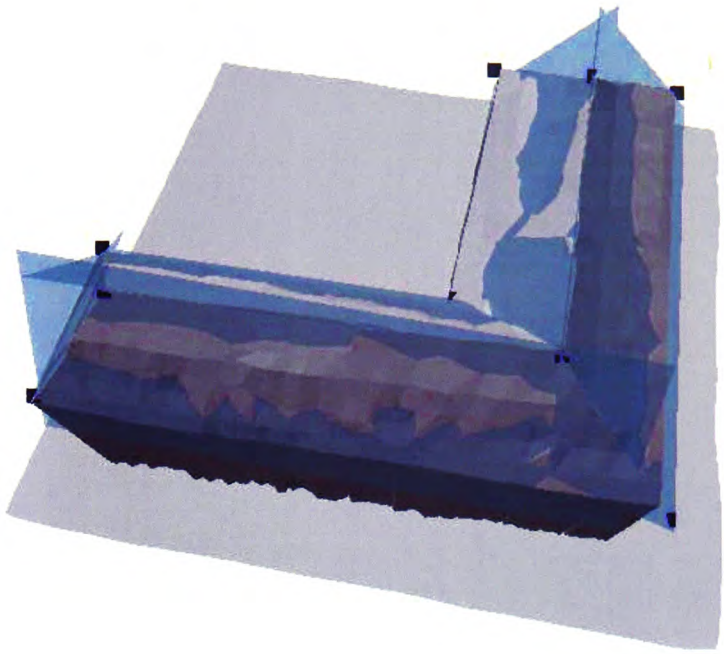


Figure 6.2: Intersection points of an L-shaped building

walls, which makes the form of the building too complicated. Fig. 6.3 shows the points added on the surface.

The steps for inserting the building outline and ridge points on the terrain include:

1. Take all the intersection points.
2. Estimate the z-value (ground value) of the points.
3. Add the points using the CDT on the terrain.

If there is an extension attached to the main building, data points inside either one of the buildings are deleted. Then the building corners and ridges are added on the terrain surface. Fig. 6.4 shows the 2D view of a T-shaped building with an extension.

6.3.3 Extrude and Reconstruct Building Using Euler Operators

When the points are added on the surface, the building is extruded to its height using the Euler Operators (described in Chapter 3). The building height is the z-value of the building corner points, which are found by the three planes intersection. Fig. 6.5 shows the extruded building with a flat roof.

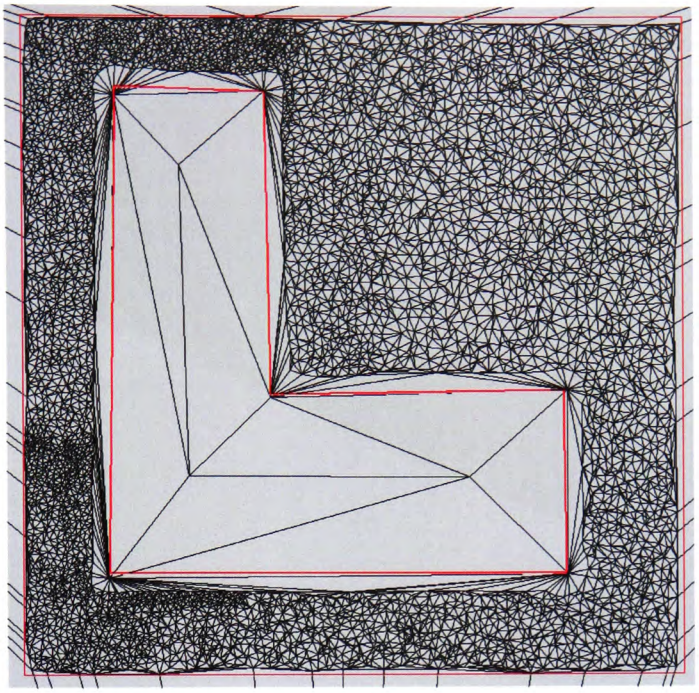


Figure 6.3: An L-shaped building's points are added on the surface

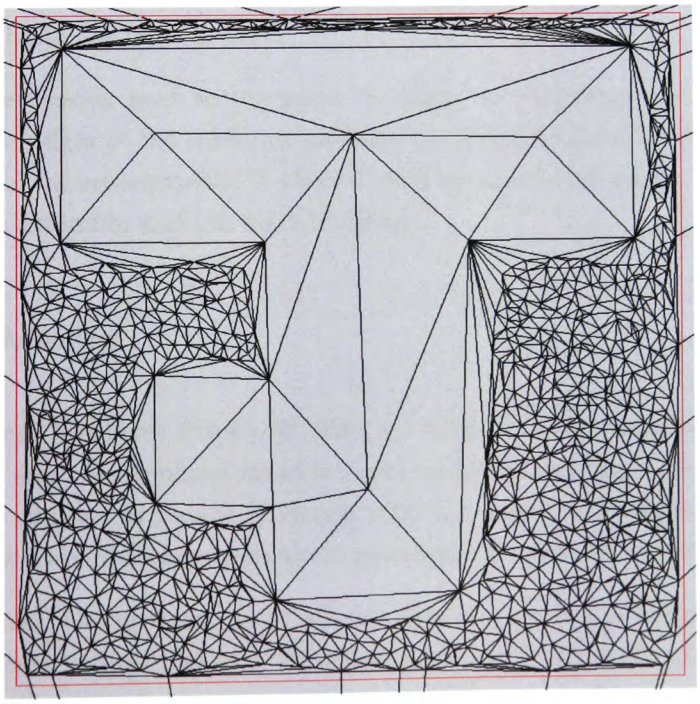


Figure 6.4: A T-shaped building's (with an extension) points are added on the surface

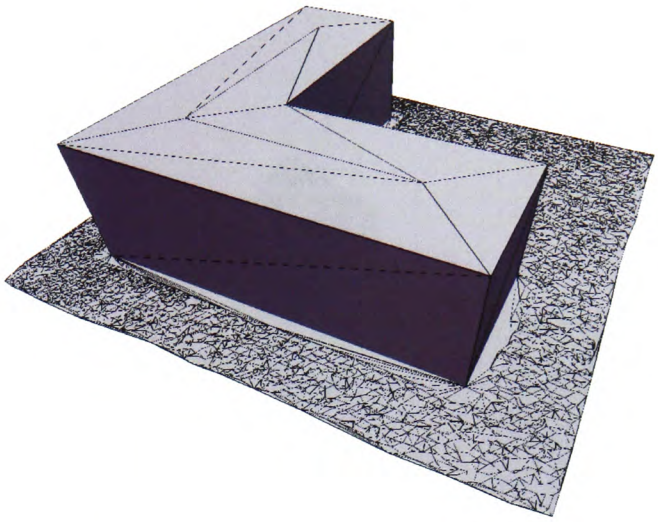


Figure 6.5: A flat-topped L-shapped building

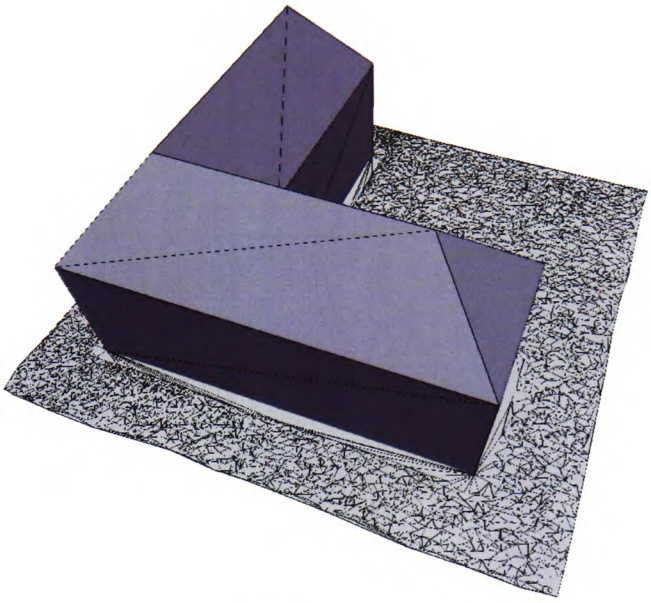
However the sample L-shaped building is not a flat roof building. The roof will have to be changed. The three planes intersection gives not only the z-values of the building corners, but also the heights of the ridge points. Therefore the hipped roof is created by uplifting the ridge points. To uplift the ridge points, the z-value is changed. Figures 6.6(a) and 6.6(b) show the final reconstructed buildings in a 3D view.

When there is an extension next to the main building, two buildings are extruded from the terrain surface. The ridges of the roofs are modified to their calculated heights. Fig. 6.7 shows two different views of the reconstructed T-shaped building and its extension. There is a common wall in between the extension and the main buildings.

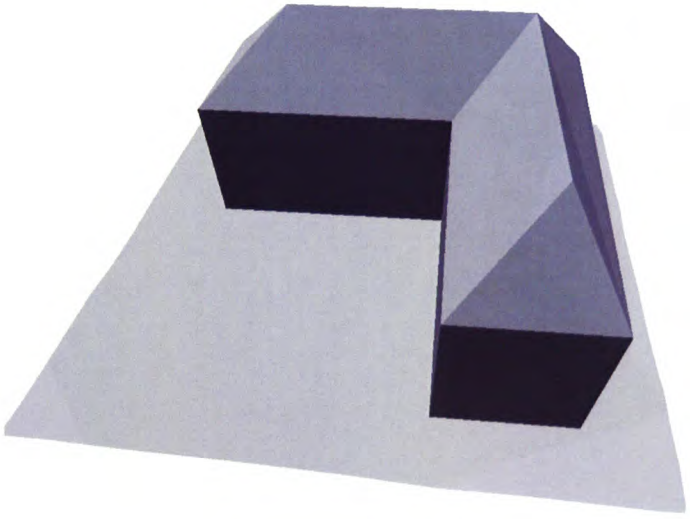
6.4 Program Manual

The proposed system has been developed using an object-oriented programming language - Delphi version 6.0, and the graphical interface uses an OpenGL library. The data file has an extension of (*.pts) which is in an ASCII format with x, y and z coordinates. Fig. 6.8 shows an example of a data file where each coordinate is separated by using an empty space.

When a file is opened, the program reads all the data and creates a Delaunay triangulation. Then the triangulation is displayed in a 2D viewing window. When the "OpenGL" button is pressed, a 3D view of the loaded terrain surface is shown in another new window. Some basic tools are essential for the user which include viewing, drawing and navigating tools.

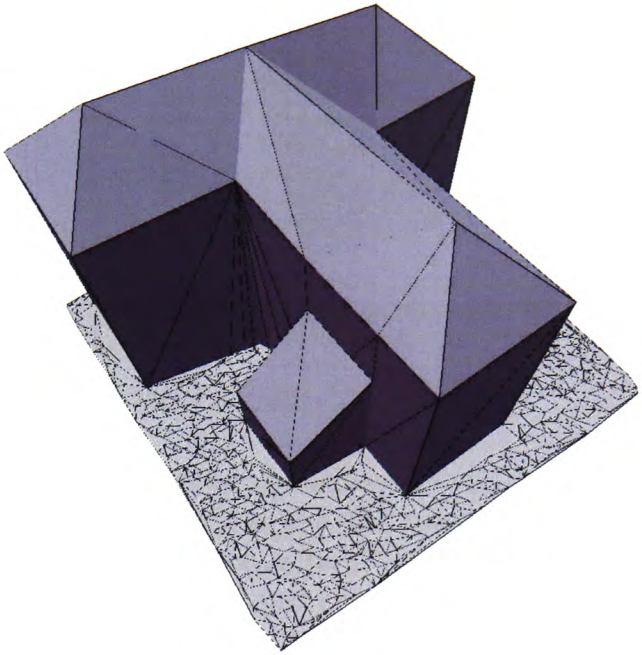


(a) View 1 with edges

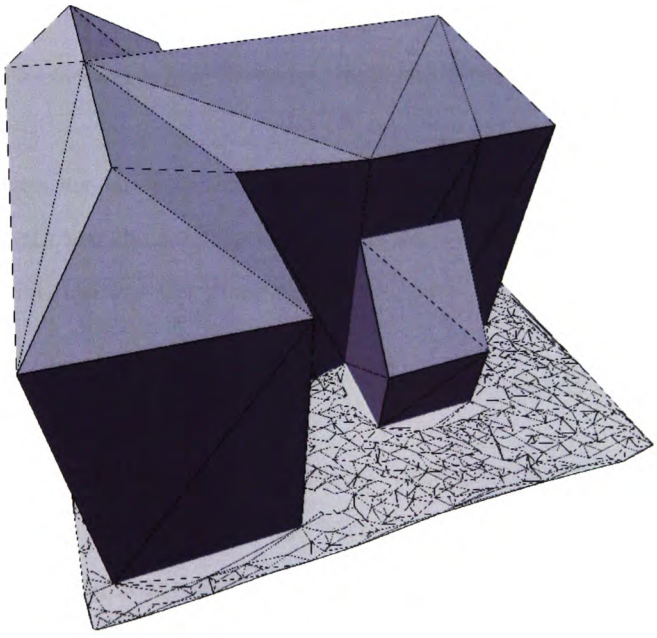


(b) View 2 without edges

Figure 6.6: Two 3D views of the L-shaped building



(a) View 1



(b) View 2

Figure 6.7: Two 3D views of the T-shaped building and its extension

```
308276.5 188876.5 62.37
308271.5 188876.5 64.42
308266.5 188876.5 68.27
308261.5 188876.5 67.26
308256.5 188876.5 67.21
308251.5 188876.5 74.68
308246.5 188876.5 77.04
308241.5 188876.5 73.14
308236.5 188876.5 68.23
308231.5 188876.5 68.34
308226.5 188876.5 68.91
308221.5 188876.5 76.75
308216.5 188876.5 75.57
308211.5 188876.5 73.20
```

Figure 6.8: An example of a data file

6.4.1 2D User Interface of the Program

Fig. 6.9 shows the 2D user interface of the program and several functions on the toolbar include importing and exporting data files, viewing ArcView Shapefile, and inserting and deleting data points. On the top left hand corner, three boxes display the x, y and z coordinates of the cursor's current position. The "Big Triangle" button creates the first big triangle to start the triangulation with three vertices, shown in Fig 6.10(a) where the data points must be inserted inside the red square. The "Big Circle" button creates the unit hemisphere by 32 vertices with no data point inside, and displays as a circle in the 2D interface in Fig. 6.10(b).

The user can choose to display eight different items in the drawing tools (Fig. 6.11(b)) check-boxes:

- **Delaunay Edges** are the edges of the triangulation as black lines.
- **Delaunay Points** are the vertices of the triangles as black dots.
- **Delaunay Altitudes** are the z-values (height) of the vertices which are shown in black digits.
- **Vitrual Circle** is the unit hemisphere drawn as a 2D cirlce in blue when the data points are added on the hemisphere.
- **Voronoi Edges** are the edges of the Voronoi diagram in red lines.
- **Voronoi Points** are the vertices of the Voronoi diagram with red dots.
- **Bldg - Building Edges** are added on the triangulation in blue lines.
- **Bldg - Building Lines** are the building outlines displayed, not added on the triangulation in thick red lines.

The five viewing tools are shown in Fig. 6.11(a) which from left to right include:

- **Zoom In** to magnify the viewing centre and enlarge the size of the object.

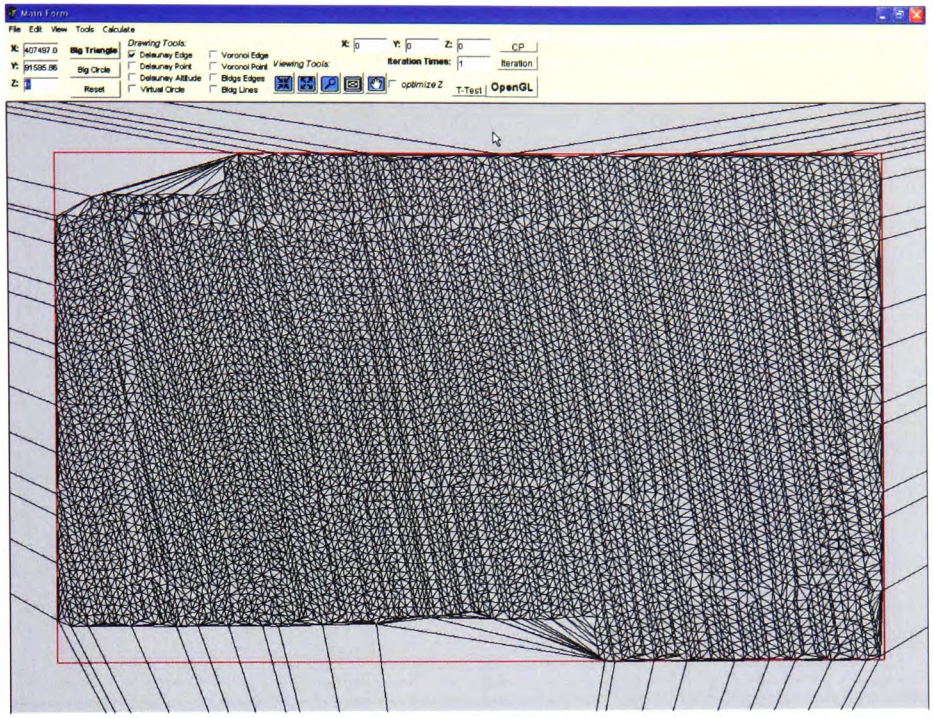
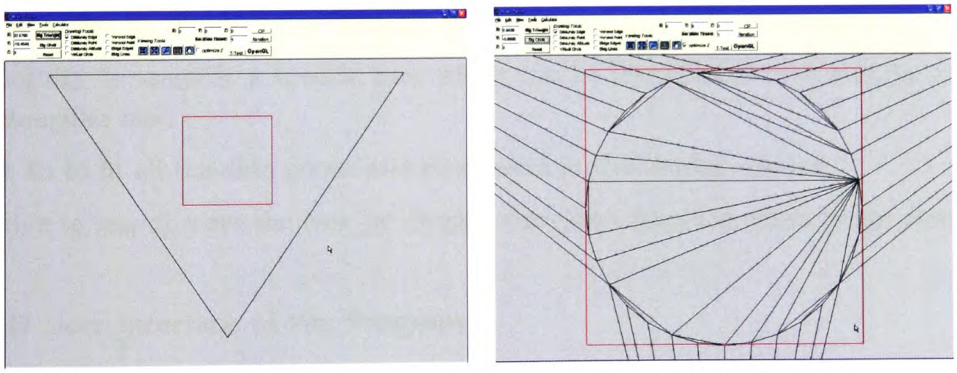
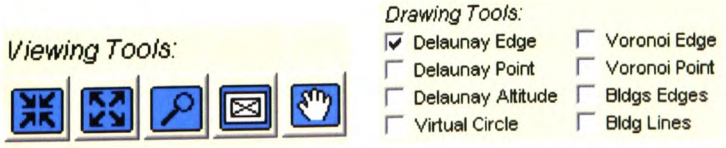


Figure 6.9: 2D user interface of the program



(a) Big triangle (b) The unit hemisphere

Figure 6.10: Initial triangle or circle of the program



(a) Viewing tools (b) Drawing tools

Figure 6.11: The tools in the 2D user interfaces

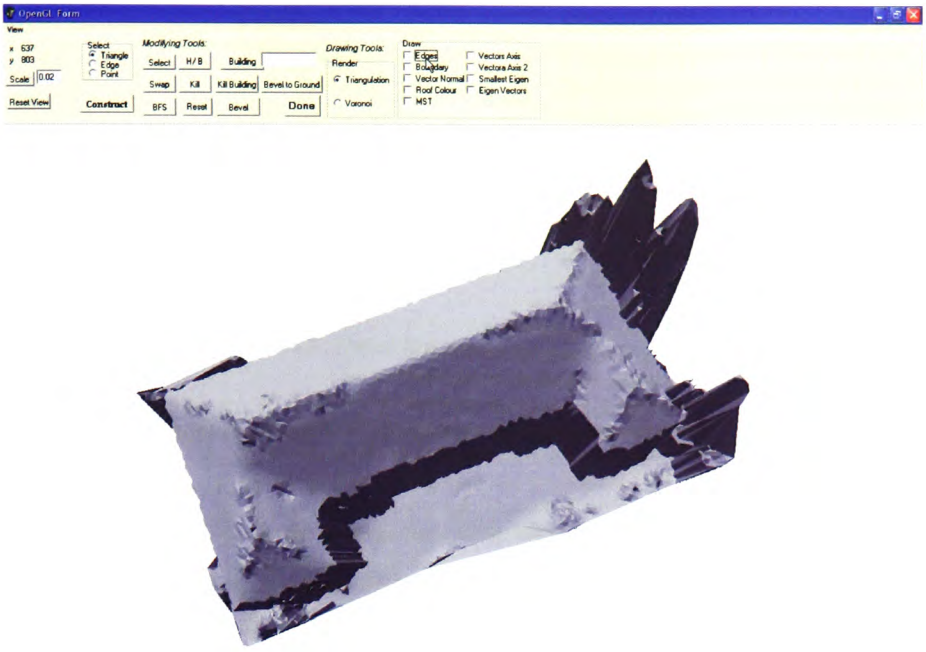


Figure 6.12: 3D user interface of the program

- **Zoom Out** to minimise the viewing centre and reduce the size of the object.
- **Magnify** to magnify a specific area which can be identified for the user by drawing a rectangular box.
- **Fit In** to fit all the data points and show them in the viewing window.
- **Move** to pan or move the view by draging the cursor from the origin to the destination.

6.4.2 3D User Interface of the Program

To display the data points in 3D, the user presses the "OpenGL" button in the 2D user interface. The 3D interface is shown in Fig. 6.12. The coordinate system in the 3D interface is not according to the data coordinate system because they were transformed. The coordinates of the data points are transformed to the 3D user interface. Two labels on the top left hand corner show the cursor's relative position. The height scale of the object can be changed because sometimes the apperance of a flat terrain surface can be improved.

Nine check boxes allow the user to display or draw different items on the surface which include:

- **Edges,** displays all the triangles edges.
- **Boundary,** shows the building boundaries which are added on the surface.

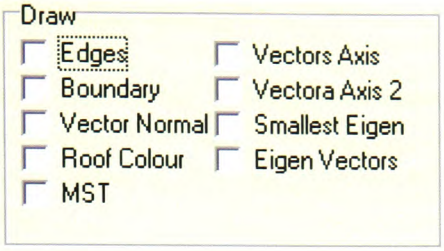


Figure 6.13: Drawing tools in 3D user interface

- **Roof Colour,** shows clustered triangles in different groups (each colour represents a group).
- **MST,** shows a minimum spanning tree from the chosen starting point in blue lines.
- **Vector Normal,** shows all the normal vectors of the triangles.
- **Smallest Eigen,** shows the eigenvector which has the smallest eigenvalue.
- **Eigen Vectors,** shows all the eigenvectors.

No navigating tools are shown on the toolbar in the 3D user interface, but they include zoom in and out, rotate right and left, and movement (pan) of the object. To navigate in the program, a wheel mouse is recommended because it will be easier for zooming and rotating the object. The user right clicks the mouse and the cursor disappears to start each operation. The user can end the operation by right clicking the mouse again and the cursor reappears. The details of the operations are:

- **Zoom Out:** Roll up the wheel of the mouse to push away or diminish the object.
- **Zoom In:** Roll down the wheel of the mouse to pull the object closer (enlarge it).
- **Rotate:** Move the mouse from left to right or roll up to down, the object on the screen is rotated according to one axis at a time.
- **Pan:** Hold down the left button of the mouse and when you move it the terrain object moves at the same time. When the object has been moved to its destination, release the left button.

The user is allowed to select triangle(s), edge(s) or point(s) on the 3D user interface. First make sure which item should be selected. To select a specific item, the user can left click on it. If more than one item must be selected, press and hold the "Shift" key on the keyboard and left click on the items. To deselect a selected item, left click on the item again.

The terrain can be rendered as a triangulation or a Voronoi diagram which can be switched by choosing in the "Render" box. Two options are available, but only one can be chosen at a time.

6.4.3 Steps for Building Block Recognition

Two different methods used to search for the building boundaries use existing 2D map data or the proposed building extraction algorithm. ArcView Shapefile is the file format for importing the 2D map data in Section 6.4.3. Before finding the building outline, the imported LIDAR data is sampled to a lower resolution.

Using 2D Map Data and LIDAR Data

The user needs to import LIDAR data points by choosing **File > Open file**. A file open dialogue box appears for the user to choose a file. After loading the data points, a 2D triangulation is created and displayed in Fig. 6.14(a). The next step is to import an ArcView shapefile, the user needs to choose **View > View Building Outlines from Shapefiles**. Then an "Import Shape File" window appears in Fig. 6.14(b) which shows the maximum and minimum coordinates of the shapefile. The user has to press "OK" to confirm the file import. The building boundaries are shown using thick red lines in Fig. 6.15.

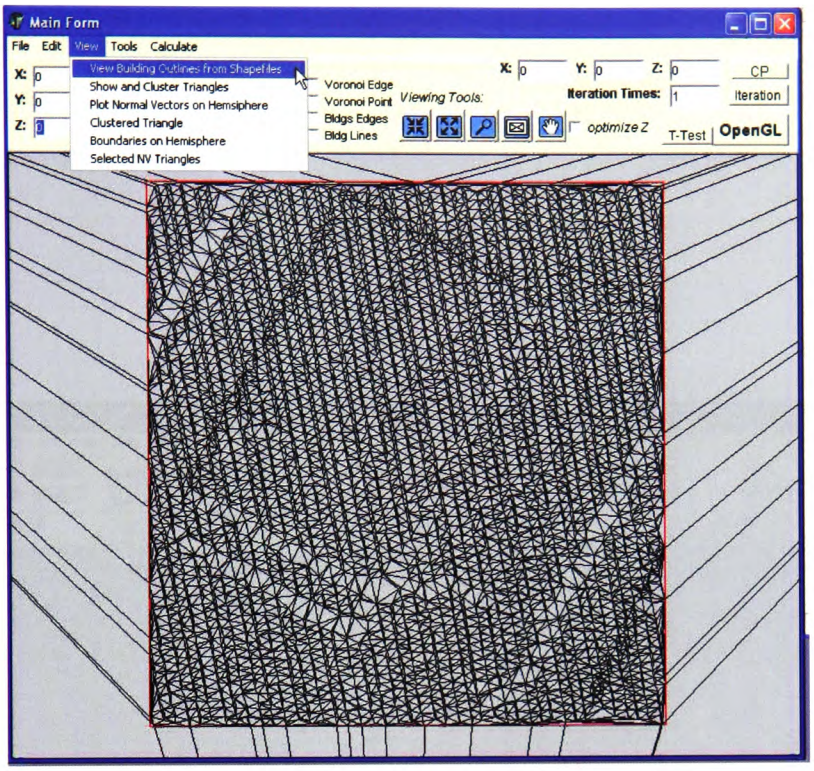
Using LIDAR Data only

When extracting a building footprint from the LIDAR data, the original data points are sampled to a lower resolution. A U-shapped building is used to demonstrate the procedure. First the user chooses **Edit > Mesh Properties** to resample the data points in Fig. 6.16(a), then a window appears for inputting a suitable disc tolerance (1.0) in Fig. 6.16(b). Fig. 6.17(a) shows the index layer.

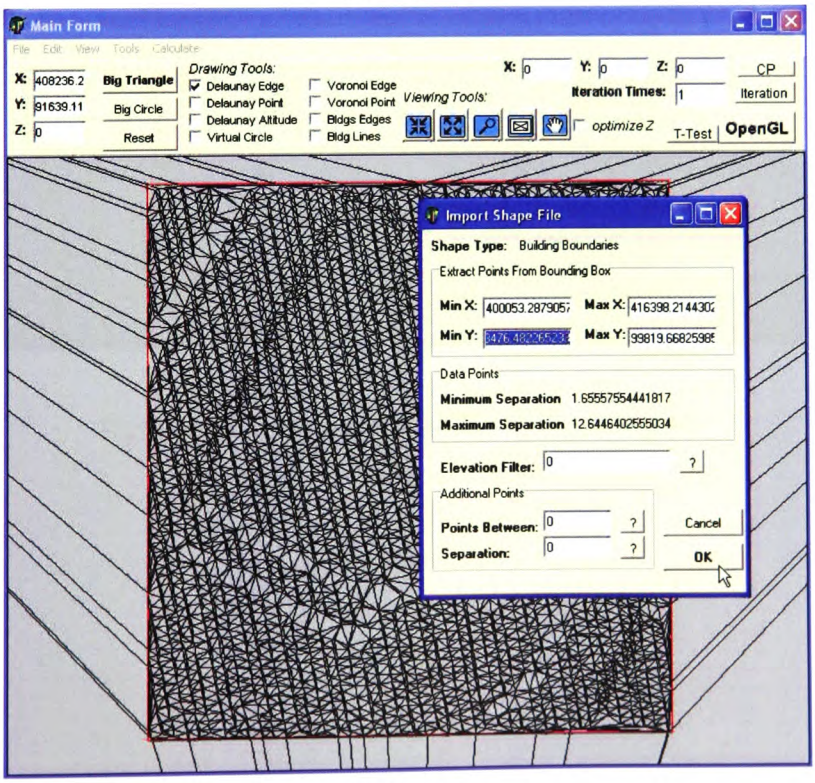
The "Reset" button removes all the data points and returns to an empty model. The next step is to import the original, and the resampled data points in Fig. 6.17(b). Two meshes are loaded into the program, and then the user presses the "T-Test" button to search for the building footprint. A confirmation window appears to check whether the result is acceptable. The user chooses "Yes" to approve the result in Fig. 6.18(a). Fig. 6.18(b) shows the building outline in thick red lines by checking the "Bldg Lines" in the *Drawing Tools* area.

6.4.4 Steps for Roof Modelling

When the building boundary is found, the next step is to search for the roof structure. All the triangles inside the building boundaries (red thick lines) are extracted in Figs. 6.19 and 6.20. Although three different clustering methods are used to group the triangles, only one step can be seen by the user. The U-shaped building is used to demonstrate the steps.



(a) Step one



(b) Step two

Figure 6.14: Steps for importing a shapefile

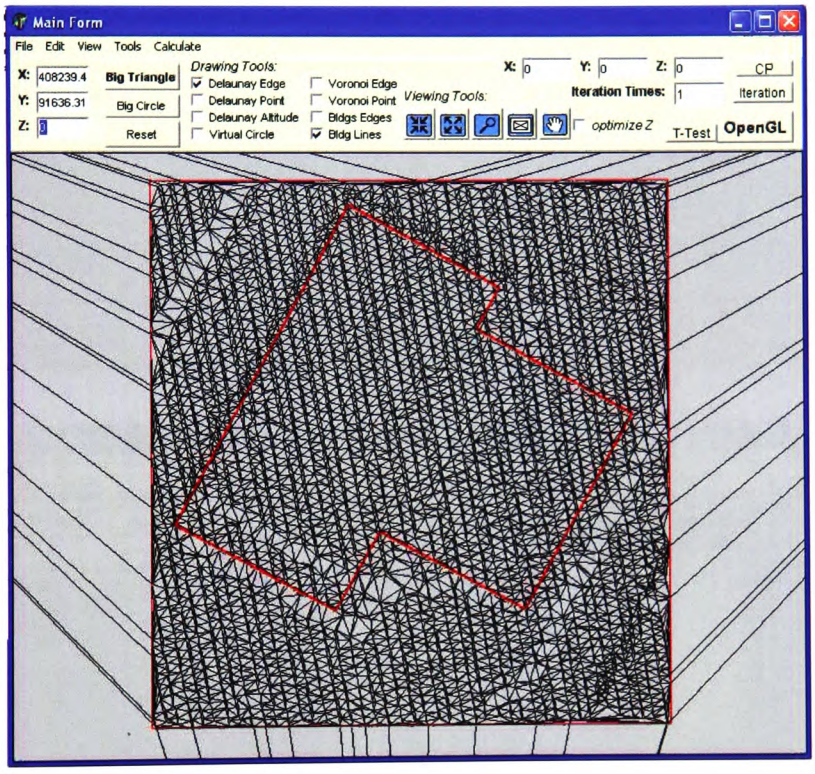
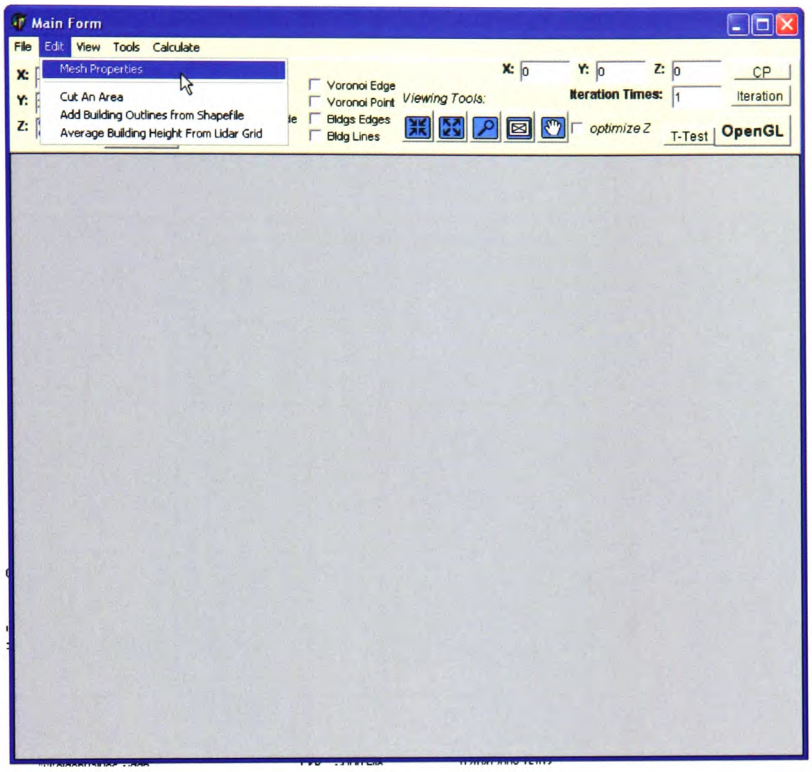
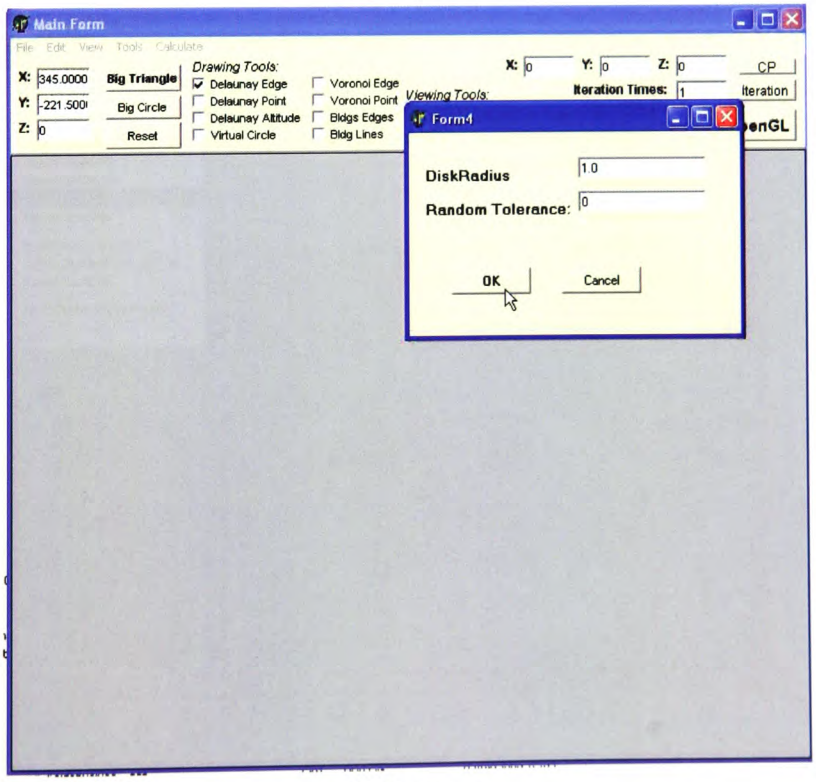


Figure 6.15: A building boundary from a shapefile

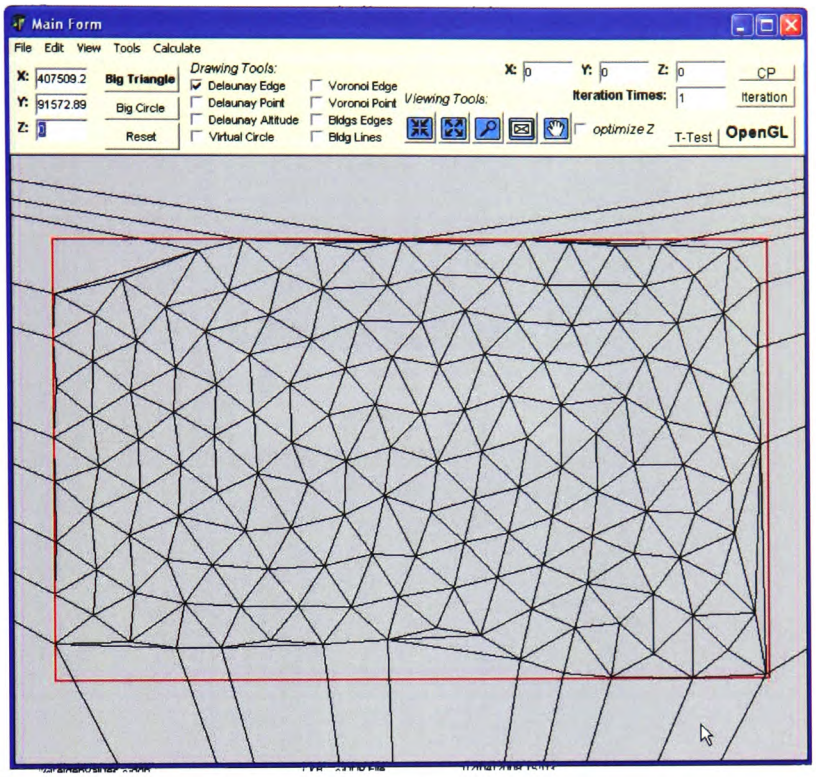


(a) Step one

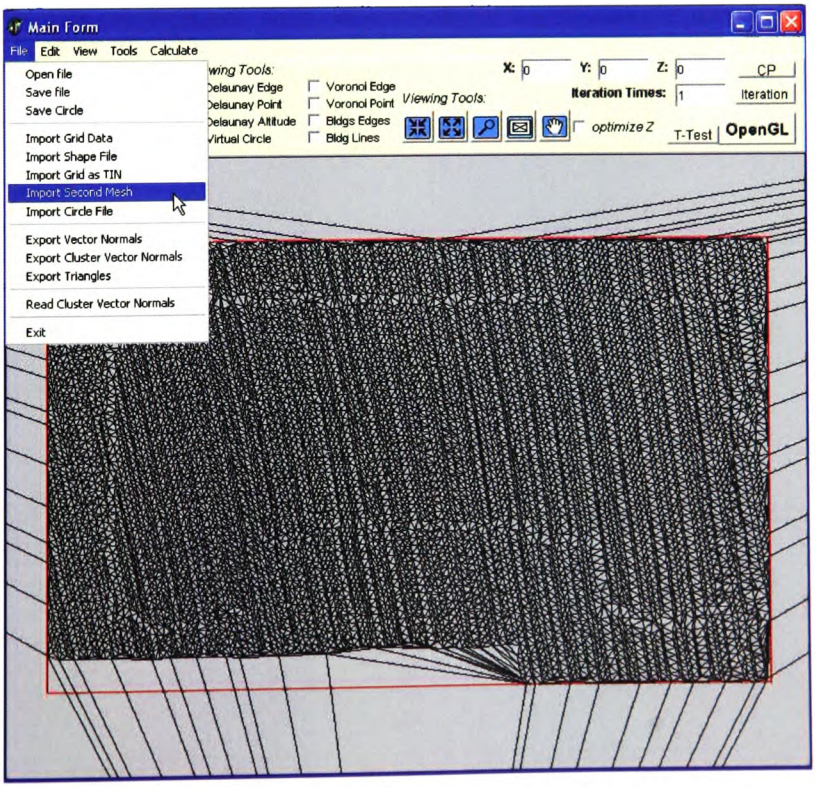


(b) Step two

Figure 6.16: Procedures (1 and 2) for building boundaries extraction

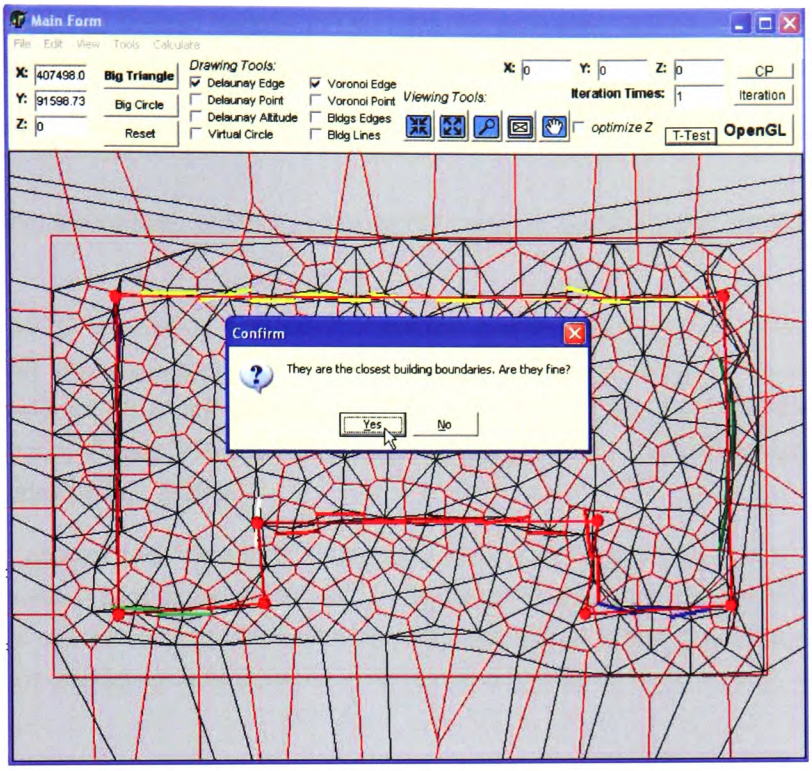


(a) Step Three

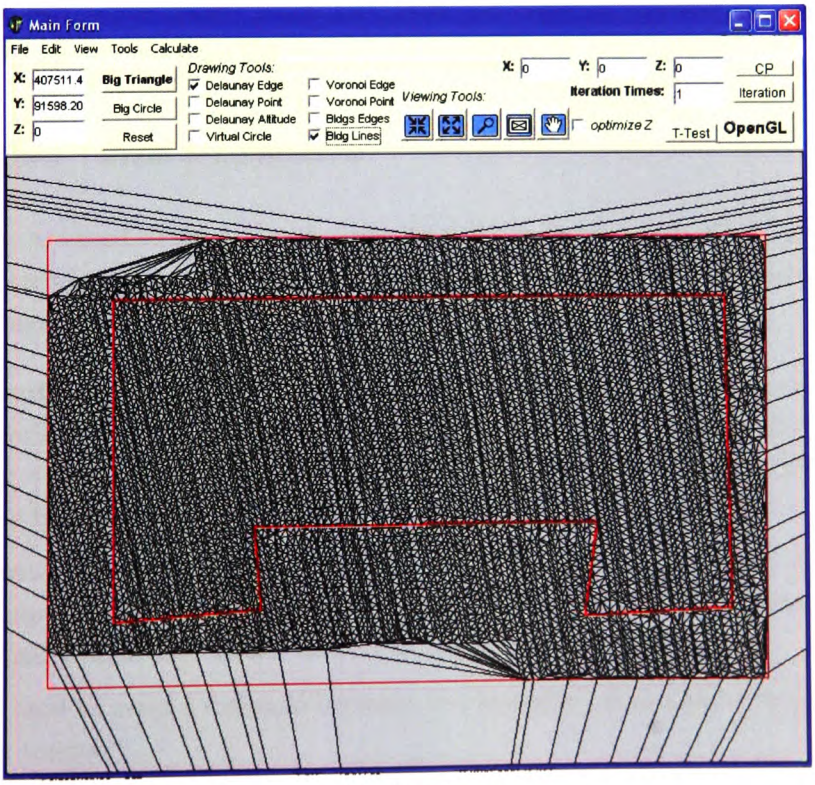


(b) Step Four

Figure 6.17: Procedures (3 and 4) for building boundaries extraction



(a) Step Five



(b) Step Six

Figure 6.18: Procedures (5 and 6) for building boundaries extraction

Fig. 6.19(b) shows the normal vectors of the triangles projected and plotted on the unit hemisphere. Then the user inputs two parameters for the orientation clustering which are:

- **No of triangles,** a minimum number of triangles to form a group and represent a roof plane.
- **Tolerance,** a minimum distance between two normal vectors so that they stay in the same group.

The "Cluster MST" button is pressed to separate the triangles and the normal vectors are clustered to four main groups with different colours in Fig. 6.19(b). Two more clustering methods are used to check on each group of normal vectors. To view the final result of all clustering methods, choose **View > Clustered Triangles** from the toolbar to show the eight groups of triangles in Fig. 6.20(a).

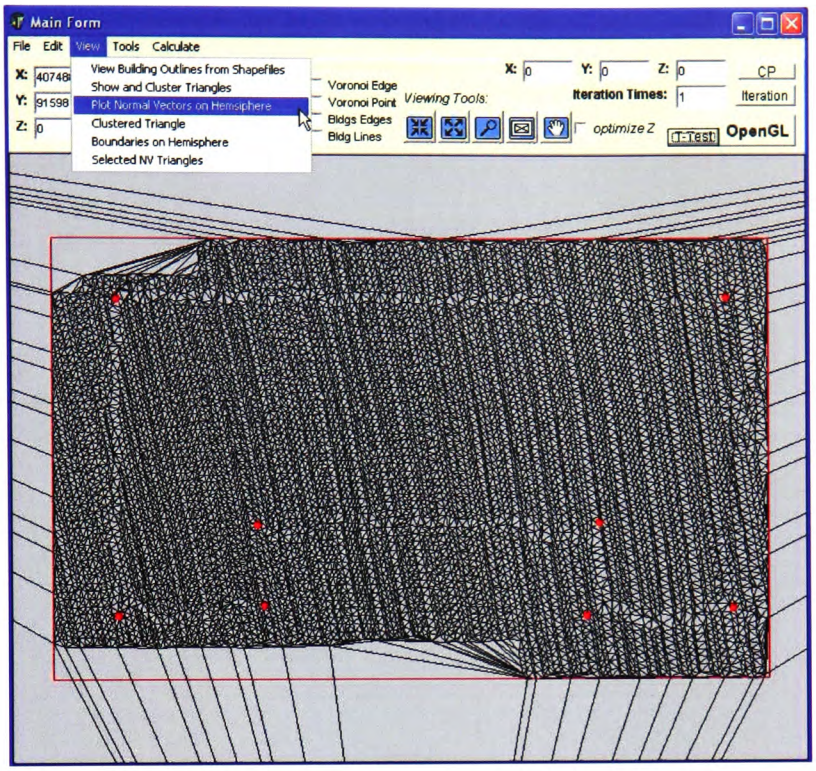
Then the user presses the "OpenGL" button to show the 3D user interface. Each group of triangles represents a roof plane which can be displayed by checking **View > Planes Perpendicular to NV** on the toolbar. Then the user chooses **View > Intersection of 3 Planes** to calculate the intersection points between every three adjacent roof planes and vertical walls in Fig. 6.20(b).

Once the user confirms the intersected points are correctly found, he or she presses the "Construct" button to remove points which are inside the building boundaries. The building is extruded and reconstructed from the terrain automatically by the program.

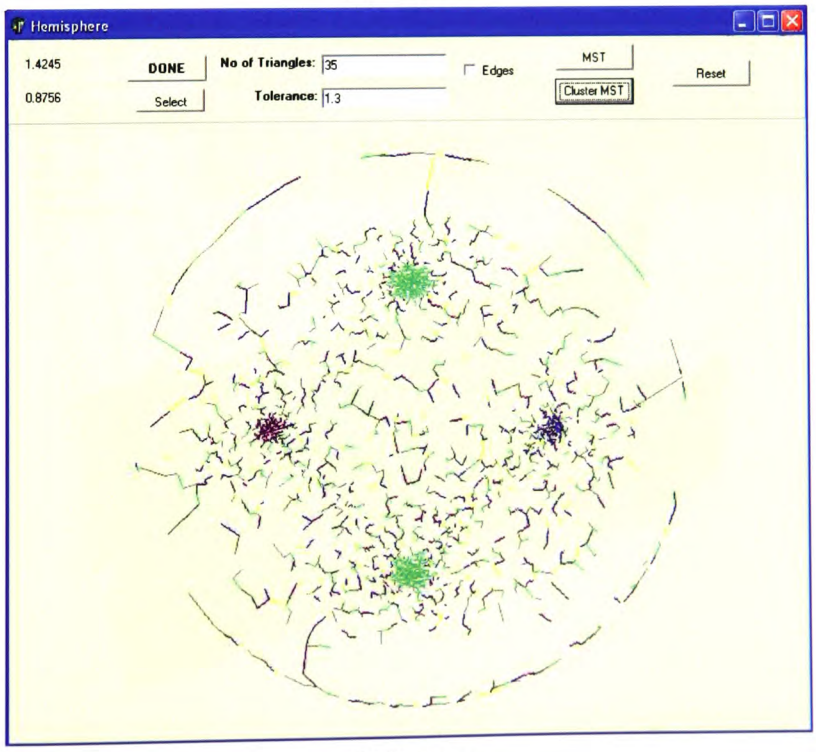
6.4.5 Real Time Terrain Modification

Euler operators are used to extrude and reconstruct the building which allows real-time modification for the terrain. Some tools available for the user for further terrain amendment include creating buildings, bridges or tunnels.

- **To extrude buildings,** press the "Building" button and then select triangles on the terrain surface and input a specific number for the height of the building (e.g. 10 = 10 meters or +10 = plus 10 meters from current height). Finally press the "Done" button to create the building.
- **H/B** means to create a bridge as in Fig. 6.21 or tunnel (hole) in Fig. 6.22. Press the "H/B" button and then select two individual triangles which are not connected to each other. Finally press the "Done" button to create the bridge or tunnel.
- **Swap** is used to swap a diagonal between two triangles which can be used to enlarge the bridge or tunnel.

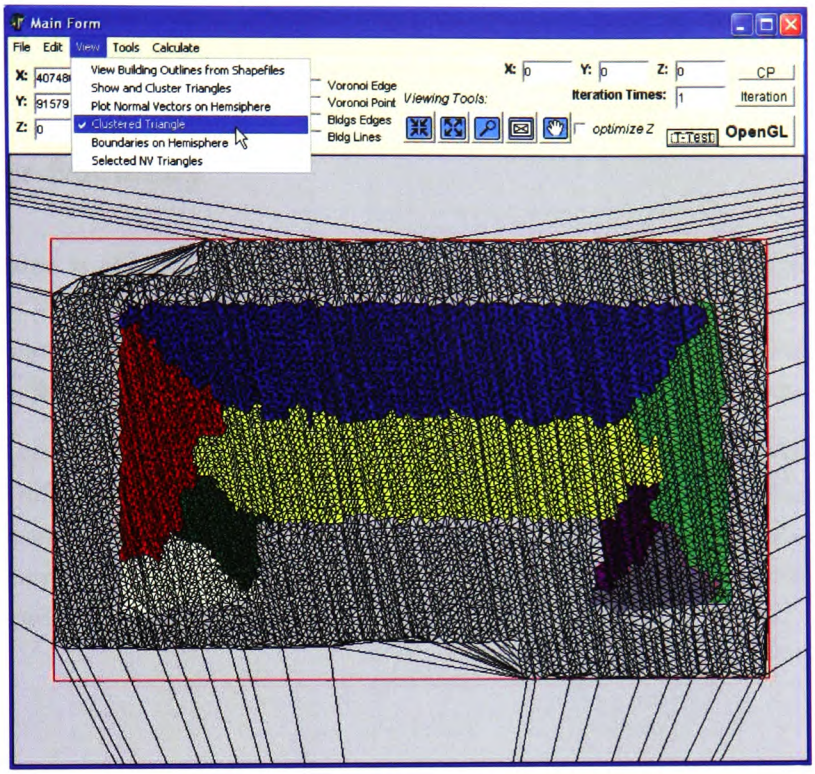


(a) Step one

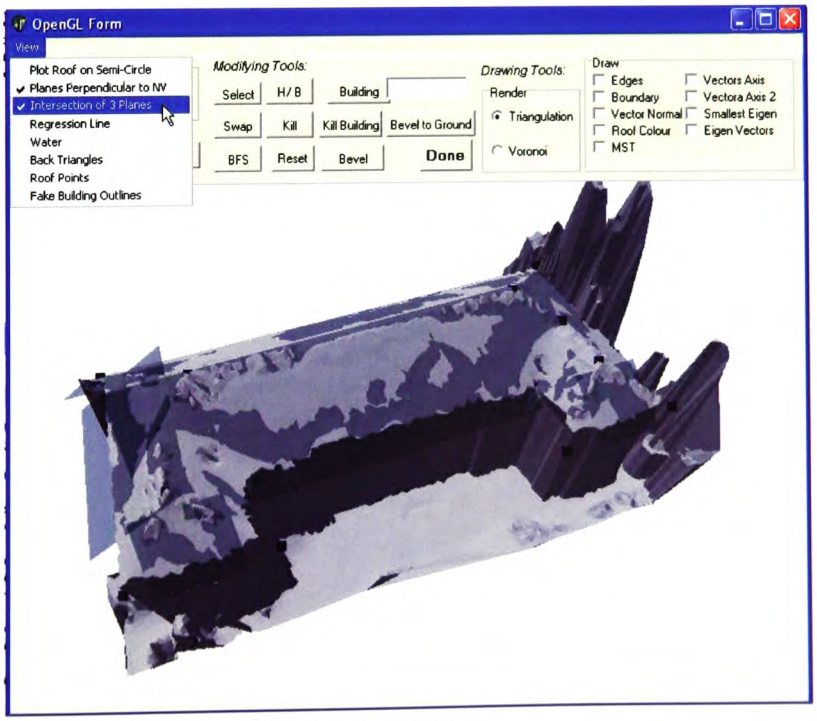


(b) Step two

Figure 6.19: Procedures (1 and 2) for roof modelling



(a) Step Three



(b) Step Four

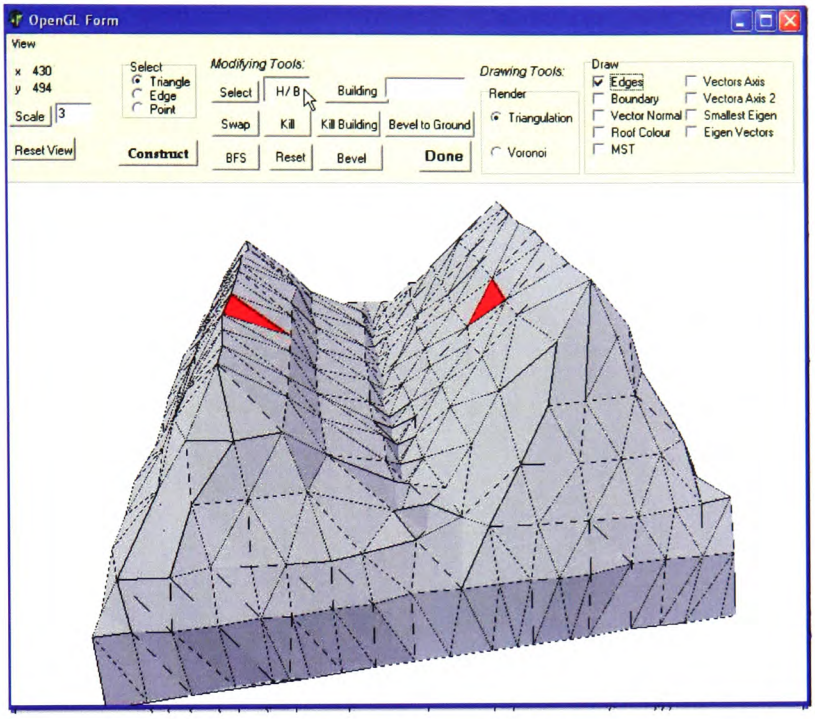
Figure 6.20: Procedures (3 and 4) for roof modelling

The created objects can be removed by selecting the object and pressing the "Kill" button. More functions are available to modify the terrain or create objects.

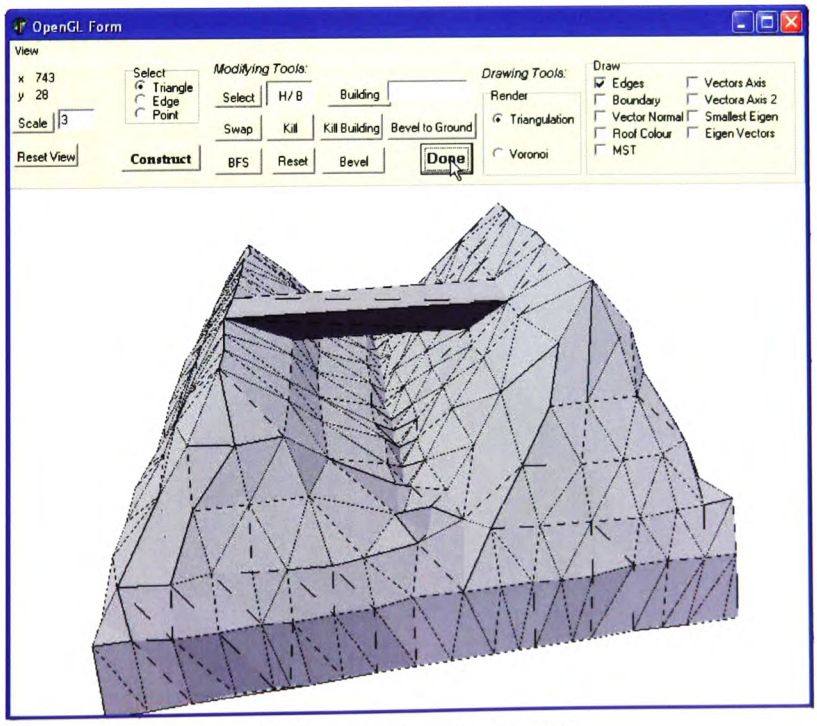
6.5 Chapter Summary

Building extrusion is the final stage to reconstructing 3D buildings from LIDAR data. The well-developed CAD tools, Euler Operators (described in Chapter 3) are used to preserve the topological connectivity when creating the 3D buildings. Beside the connected topology, the construction toolkit is easy to implement which is good for further development.

The proposed system has been developed using an object-oriented programming language - Delphi version 6.0. Section 6.4 shows some main functions of the program which include how to display the terrain objects in 2D and 3D views, extract building boundaries, search for the roof planes and modify the terrain surface. The procedures of building reconstruction are shown step by step; however it can be running automatically.

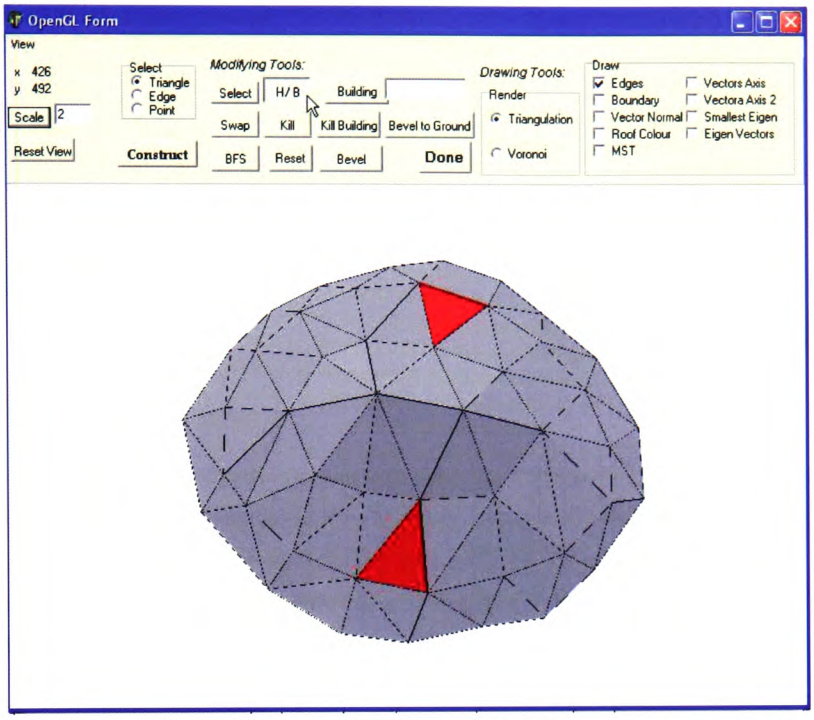


(a) Step 1 - select two triangles

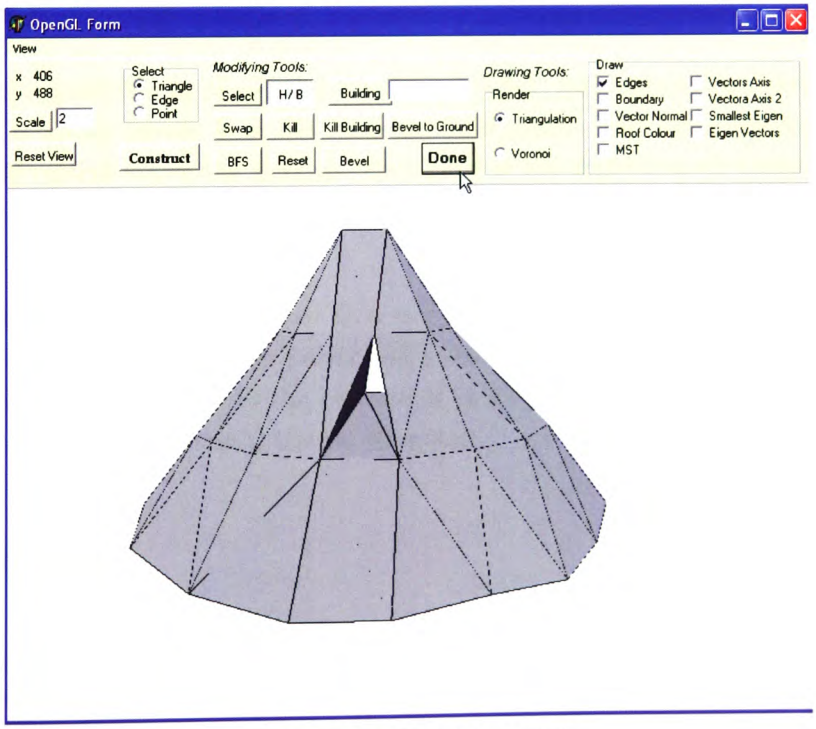


(b) Step 2 - a bridge is created

Figure 6.21: Create a bridge between two hills



(a) Step 1 - select two triangles



(b) Step 2 - a tunnel is created

Figure 6.22: Create a tunnel through a hill

Chapter 7

System Evaluation

Real LIDAR data is used to evaluate the system and the technical background shows the details of the real LIDAR data. Then demonstrations of the two proposed methods using real data are shown in sections 7.3 and 7.5. Topographical and LIDAR data are used to test the system in section 7.6. Section 7.7 evaluates the system and suggests some improvements. A summary of the chapter is given in the last section.

7.1 Technical Background

The LIDAR data, provided by the Ordnance Survey (UK), was captured by Network Mapping Limited which is under licence from the National Grid group. The flying height of the survey aircraft was 300 meters above the ground. The speed of the flight was 45 knots and the angle was 12 degrees (full scan was equal 24 degrees). The total number of flight lines was 21 which included the cross-flight lines.

The data points provided were captured at a rate of 6 million points per route-mile of survey. The operating parameters are 11 first points and 11 last points per squaremetrer of transmission survey. This point density ensures the capture of precise details of the ground objects. The captured area was in Bournemouth, United Kingdom.

7.2 Introduction

Roof modelling and building extraction are combining two methods to solve 3D buildings reconstruction problems; however artificial data were used to illustrate the ideas in the previous chapters. Most of the time these data tended to return a correct result; therefore using real data is necessary to check and evaluate the system. Two proposed methods are included in the reconstruction processes: roof modelling and building outline extractions. They are evaluated in sections 7.3 and 7.5 separately. Section 7.6 demonstrates the whole building reconstruction

system. Section 7.7 describes the difficulties and problems of the system, and suggests some solution for further development.

7.3 Demonstrations of Roof Modelling Methods

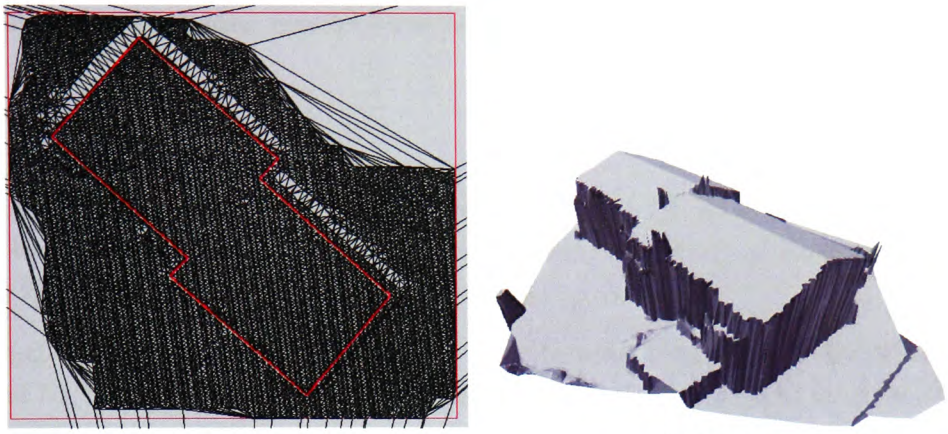
Roof modelling reconstructs the roof structure by searching for roof planes. To search for and create the roof planes, points inside the building boundaries are extracted. A Delaunay triangulation is created using the extracted points. Three clustering methods are used to separate and group the triangles which share the same properties (for example, orientation and location). Each group of triangles represents a piece of the roof plane. Three different roof styles were used to demonstrate the system. The Ordnance Survey OS Mastermap Topography Layer data are used as the building boundaries in each example. These are the red lines in each of the 2D views.

The first clustering method requires two tolerances, a minimum number of triangles to form a group and the closest distance between two normal vectors to cluster in the same group. The tolerance in the first clustering method is flexible and can be input by the user. The second clustering method projects the centre points of the clustered triangles with the same orientation onto an averaged normal vector (described in chapter 5) and the tolerance is 2 metres. The tolerance is set as a fixed value of 2 meters for the third clustering method, which clusters the triangles according to their geographic location (in chapter 5). If the triangles in the same group are too far from each other, they will be separated into two groups. The length tolerance is set to be 2 meters because normally a storey is around 2 - 3 meters high.

7.3.1 Simple Two Planes Building

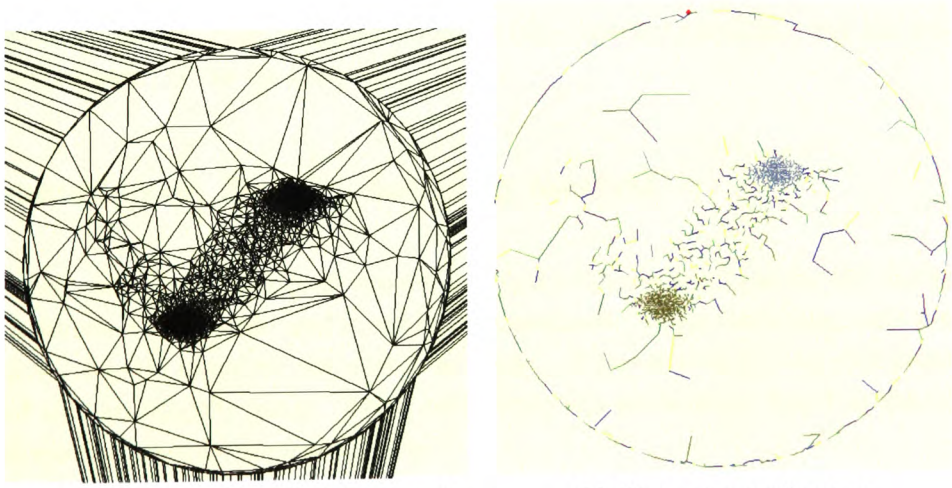
An example of a two planes roofed building is used to illustrate the roof remodelling algorithm. Data points inside the building boundaries are extracted to create a Delaunay triangulation in Fig. 7.1. The next step is to extract all the normal vectors from the triangles and perform the three clustering methods.

The first clustering method separates the triangles according to their orientations. A minimum spanning tree (MST) is created (in Fig. 7.2(a)) using the plotted normal vectors. The normal vectors are clustered according to their distances on the hemisphere. If the normal vectors are further apart than 1.2 meters, they will fall in two different groups. The minimum number of triangles in each group is set to be 40. Fig. 7.2(b) shows two groups of normal vectors on the hemisphere. Then the other two clustering methods are used to separate the clustered triangles.



(a) 2D view (b) 3D view

Figure 7.1: An example of a simple two plane roofed building



(a) Normal vectors are connected by a MST (b) Clustered normal vectors

Figure 7.2: Plotted normal vectors on the unit hemisphere

The example contains two buildings which are connected together. Three groups of triangles are found to form three roof planes 7.3(a). Having three clusters instead of four is because two roof planes connect to each other (the blue triangles in Fig. 7.3(b)). The red and purple triangles do not connect to each other; therefore two roof planes are formed.

7.3.2 T-shaped Building

An example of a T-shaped building contains five roof planes. Data points within the building boundaries are extracted and used to create a Delaunay triangulation. Figures 7.4(a) and 7.4(b) show the 2D and 3D views of the T-shaped building.

The normal vectors of the triangles are extracted and plotted onto the unit hemisphere. Figures 7.5(a) and 7.5(b) show that the normal vectors are connected and clustered using an MST. After the three clustering methods, five roof planes are found.

Region growing is used to find the relationships between the roof planes. Figures 7.6 and 7.7 show the differences before and after region growing in 2D and 3D views.

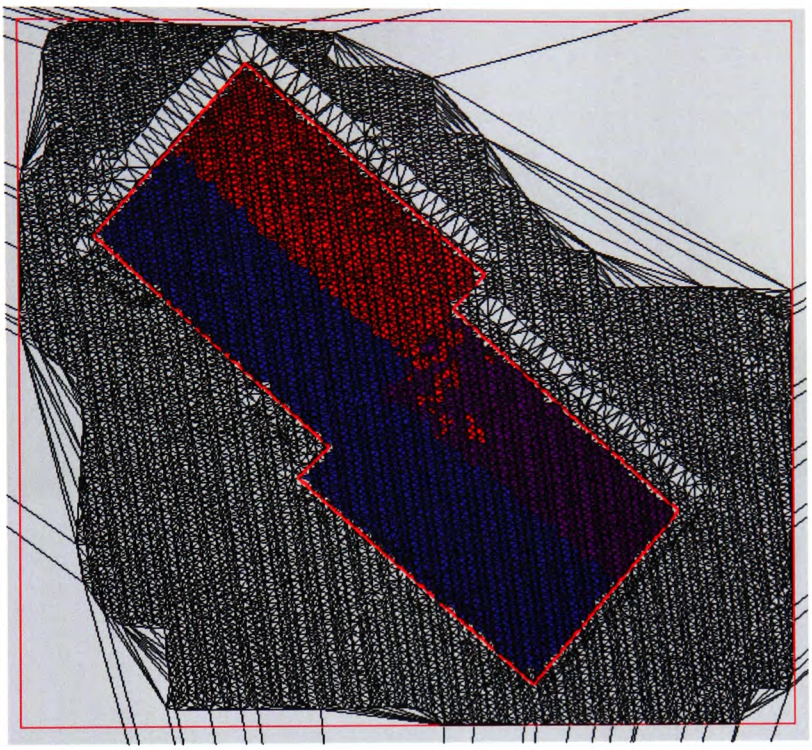
7.3.3 L-shaped Building

An L-shaped building is used to demonstrate the roof remodelling method. Fig. 7.8 shows the sample building in 2D and 3D views. When the clustering methods are processed, four roof planes are found from the L-shaped building (Fig. 7.9(a)). Fig. 7.9 shows the roof planes (the blue transparent planes), the clustered triangles (the coloured triangles) and the intersection points (blue squares) of the roof planes.

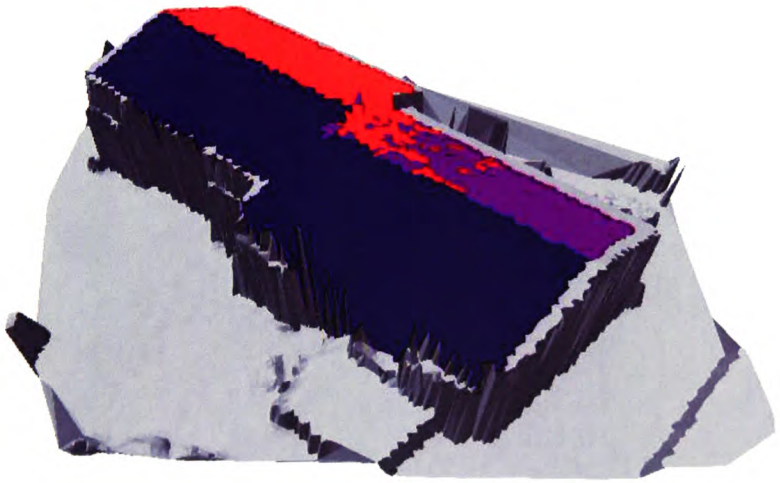
7.4 Evaluation on Roof Remodelling Method

The proposed roof remodelling methods show a good result on searching for the roof planes. In the first example, a two planes roof is used to demonstrate. At the beginning, only two clusters of normal vectors are found on the unit hemisphere. However actually the roof planes on the right-hand side are slightly apart. Three roof planes are successfully found at the end which shows the method gives an accurate result.

Reliable results are shown from the second and third examples. A T-shaped building is used to demonstrate the method in the second example and five roof planes are found. Though only four clusters are found on the unit hemisphere in Fig. 7.5, two roof planes are facing the same direction. The clustering algorithms successfully find the five roof planes at the end. In the

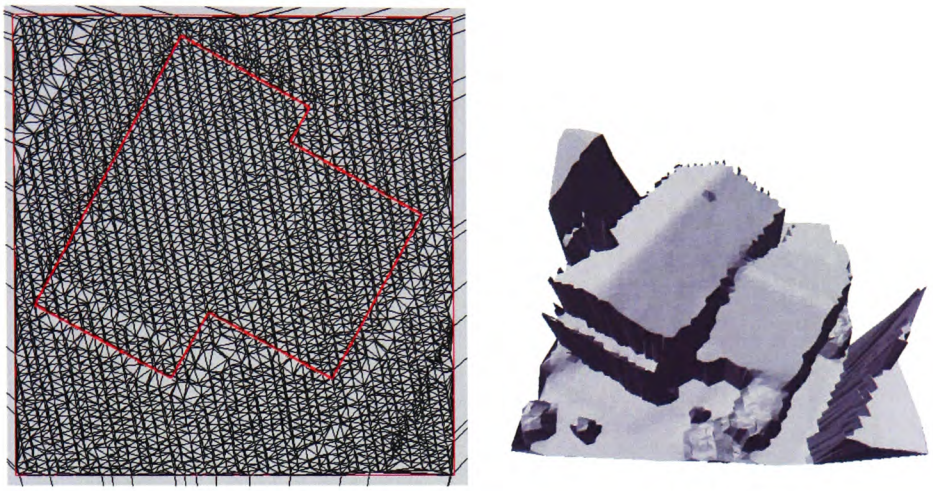


(a) 2D view



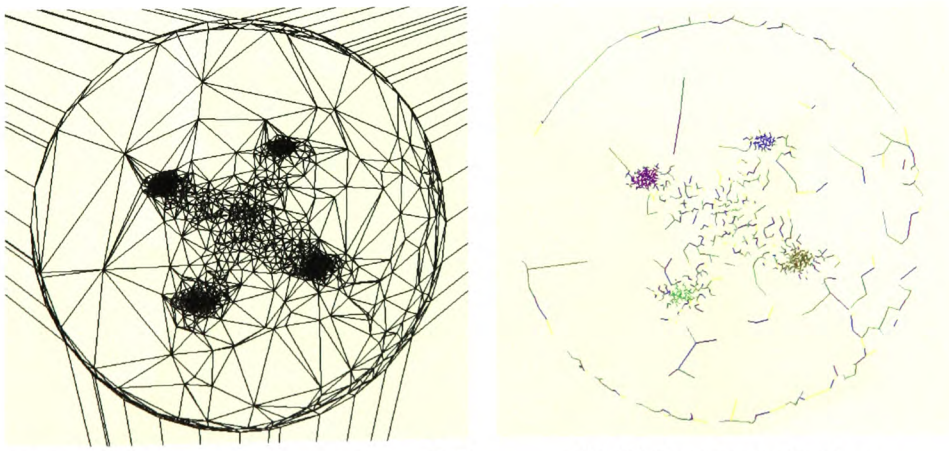
(b) 3D view

Figure 7.3: The clustered triangles of a 2 plane roofed building



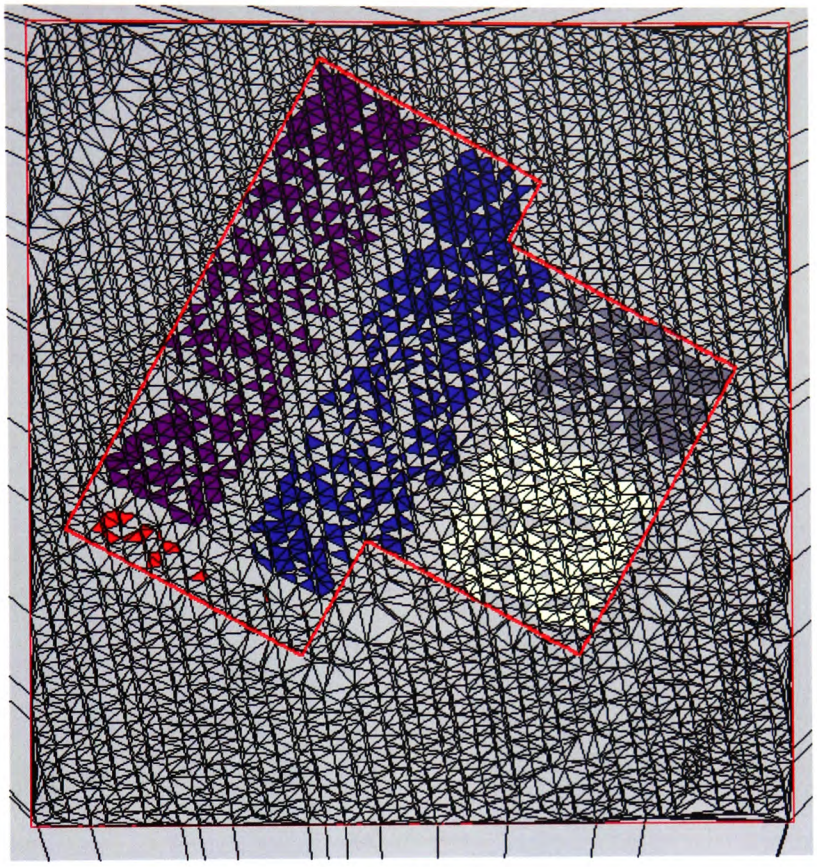
(a) 2D view (b) 3D view

Figure 7.4: An example of a T-shaped building

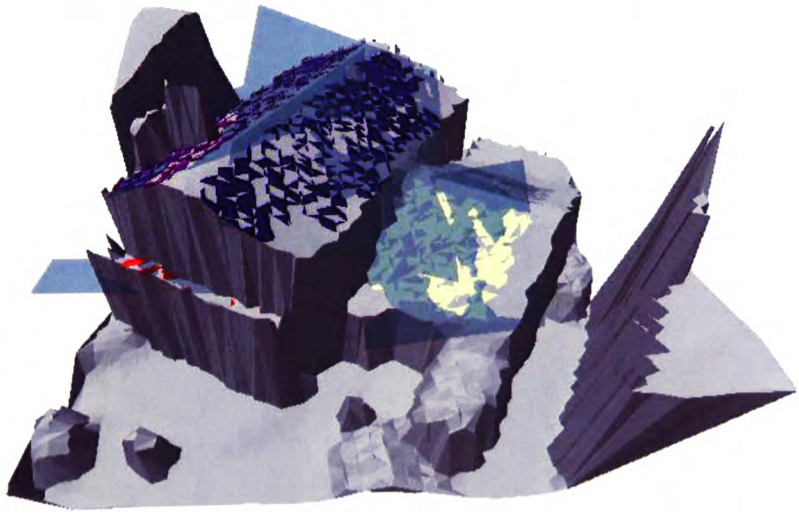


(a) Normal vectors are connected by an MST (b) Clustered normal vectors

Figure 7.5: Normal vectors on the unit hemisphere

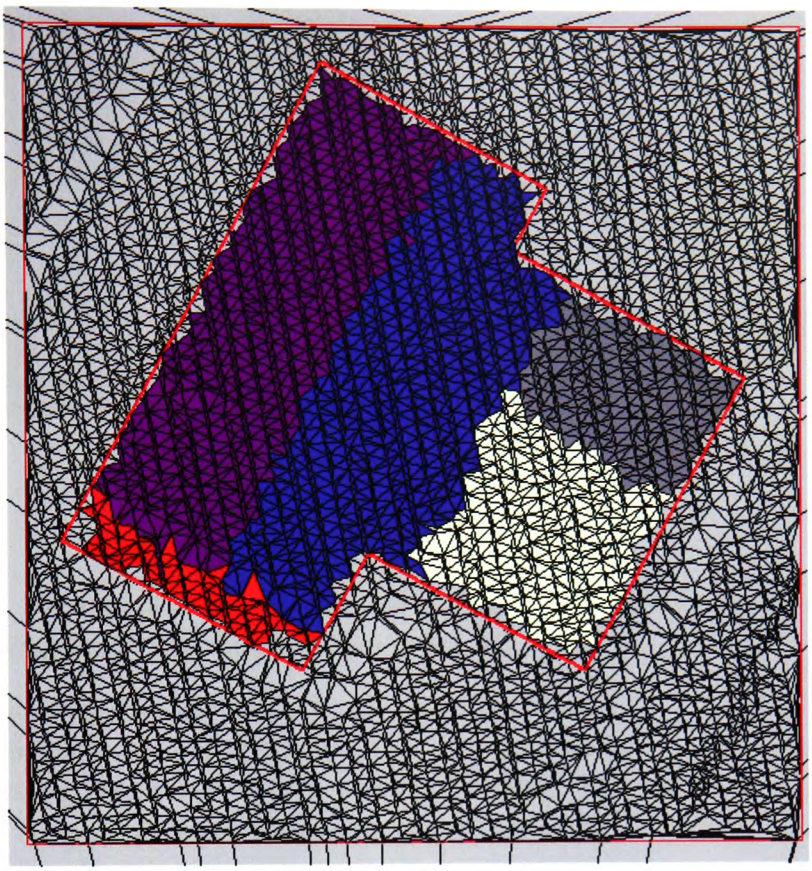


(a) 2D view

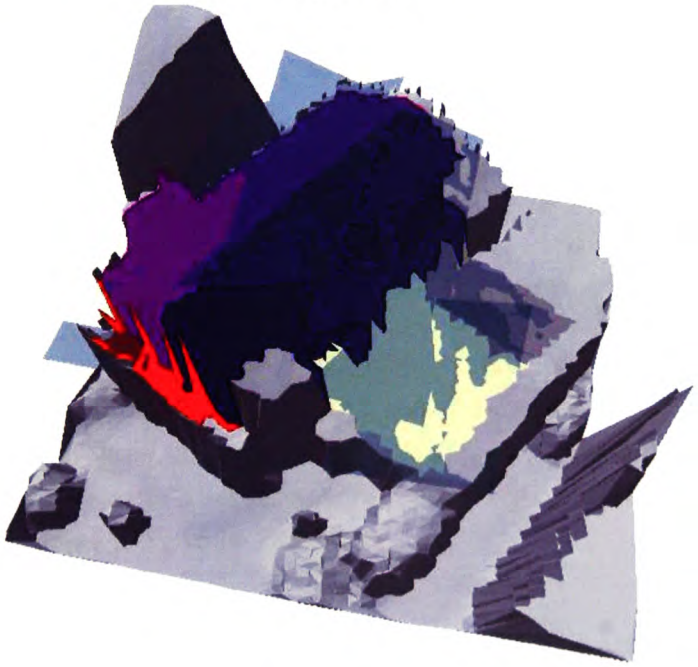


(b) 3D view

Figure 7.6: Clustered triangles of the T-shaped building

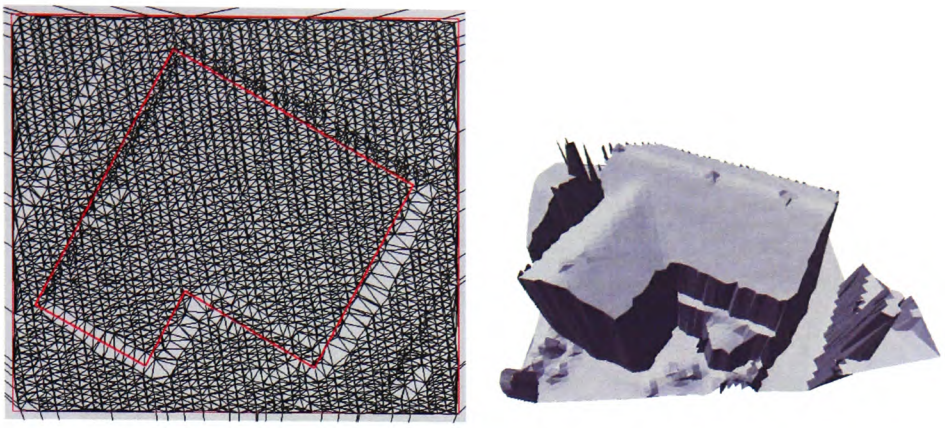


(a) 2D view



(b) 3D view

Figure 7.7: Clustered triangles of the T-shaped building after region growing



(a) 2D view (b) 3D view

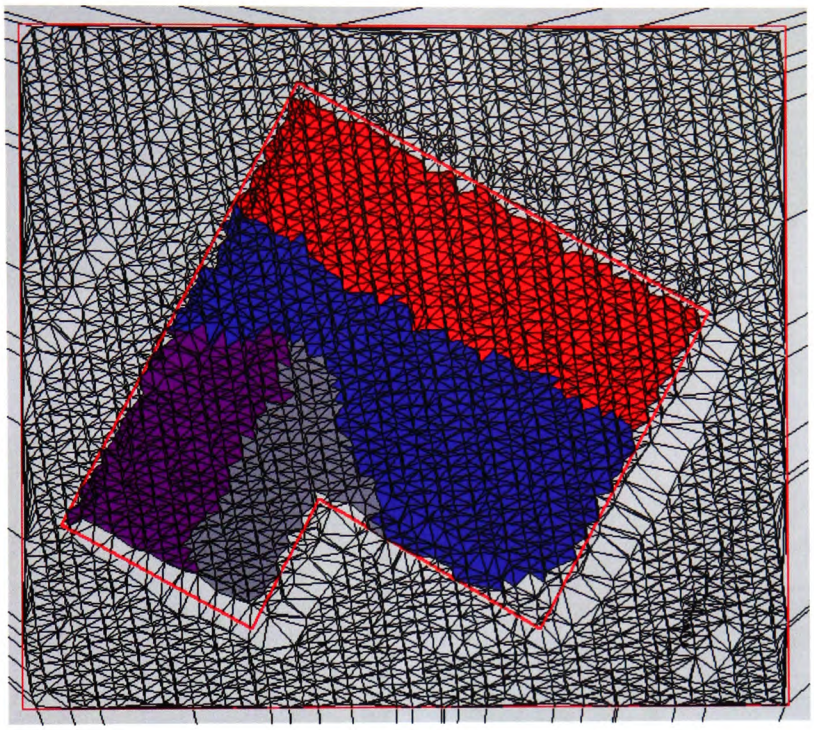
Figure 7.8: An example of an L-shaped building

third example, four roof planes are correctly found and the intersected points are calculated to form the building corners and the roof ridges in Fig. 7.9.

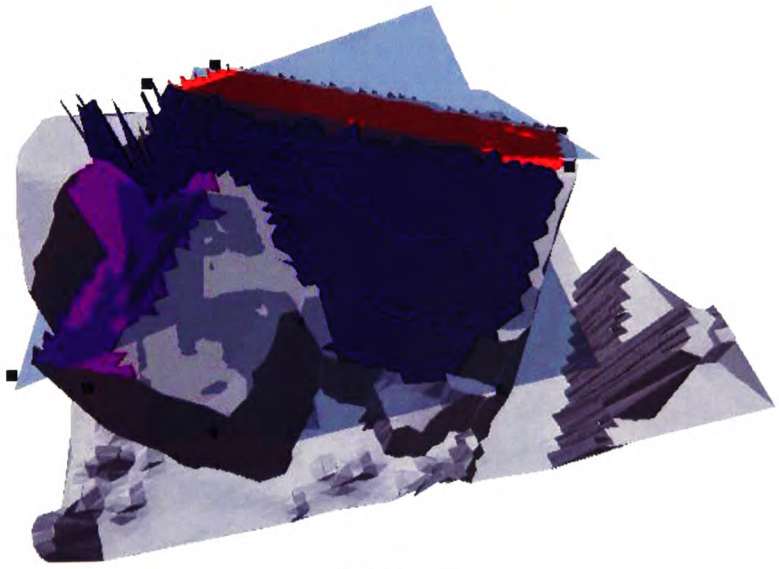
The proposed method gives some reliable and succesful demonstrations, and the underlying three clustering methods can be used separately for different applications. Many tolerances have to be set in the clustering algorithms which increase the flexibility of the input data, but decrease the automation of the system. Further development can be done to minisie the input of tolerances for the proposed method.

7.5 Demonstration of Building Boundaries Extraction

Raw LIDAR data are resampled into a lower resolution index layer to search for vertical wall segments of a building. The wall segments found are connected to form a closed loop. Then the wall segments are clustered according to their orientation and location. An average line is calculated to represent each group of wall segments. The average lines intersect to form the building corners. This is described in chapter 4. The resolution of the index layer can affect the results of searching for the vertical wall segments. If the index cell is too big, the building corners may not be found. However if the index cell is too small, it may not contain low and high points in the same cell for identifying the wall segment. The resolution of the index layers is set to achieve the best result. A tolerance is required to separate high and low areas, and the minimum height difference is set to be 2 meters.

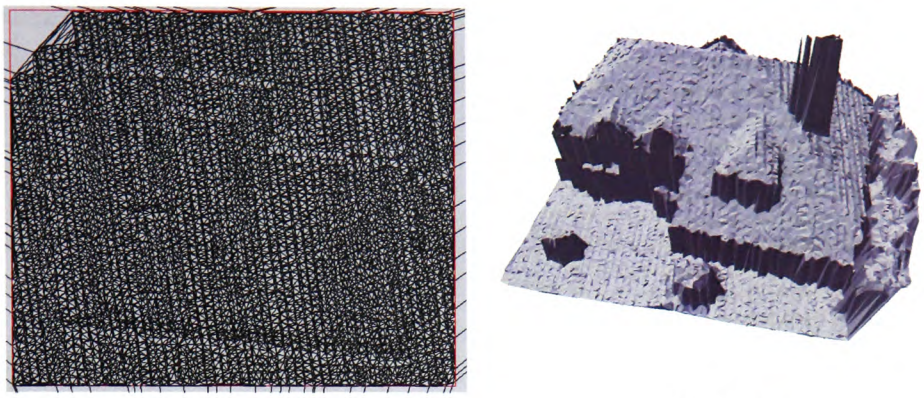


(a) 2D view



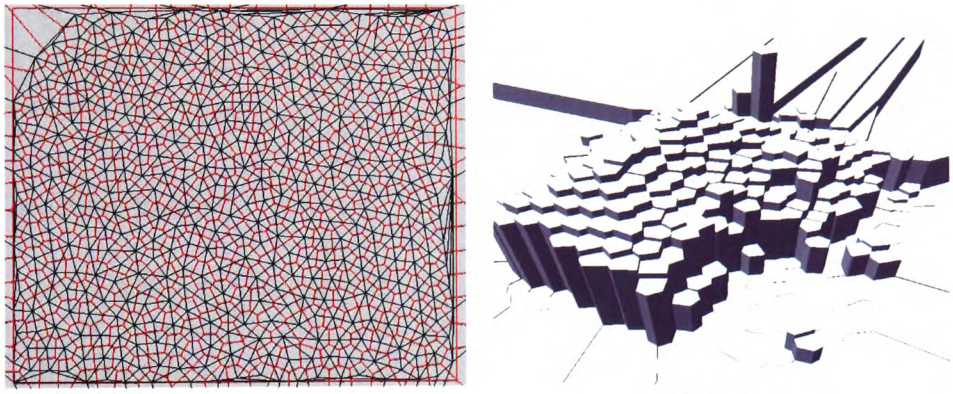
(b) 3D view

Figure 7.9: An example of a L-shaped building(clustered)



(a) 2D view (b) 3D view

Figure 7.10: An example of a L-shaped building



(a) 2D view (b) 3D view

Figure 7.11: The index layer of the L-shaped building

7.5.1 L-shaped Building

In this example, a disc of 0.4 metre is set to create the index layer. Figures 7.10 and 7.11 show the 2D and 3D views of the sample L-shaped building and the index layer of the building.

Fig. 7.12(a) shows the vertical wall segments of the L-shaped building. Real data are more complicated than artifical data (e.g. noise or unexpected errors); therefore it is not easy to connect the wall segments and form a closed boundary. The second building boundaries searching method is used. The first step of the second method is to cluster the segments according to their orientation and then cluster them by geographic location (Fig. 7.12(b)). The third step is to confirm that it is possible to form a building (described in Chapter 4). However in this case, immediately adjacent to the building there is a higer area, for example vegetation (refer to 7.10(b)). Part of the wall segments cannot be found; therefore no building outline can be located. Fig. 7.13 shows two different views of the split Voronoi cells in the 3D view. As a

result, building reconstruction could not be completed. Future work could attempt to handle these difficult cases.

7.5.2 L-Shaped Building 2

An L-shaped building 2 shows a better example of building boundary extraction because there is no high object next to it. Fig. 7.14 shows the 2D and 3D views of the L-shaped building. The raw LIDAR data is resampled to a lower resolution to form an index layer. A disc of 0.6 is used to form the index layer (Fig. 7.15).

The split vertical wall segments are extracted and clustered (described in chapter 4). Figures 7.16(a) and 7.16(b) show the six groups of wall segments found and represented using six different colours (Fig. 7.16(b)). The best fit line is calculated to represent each group of the clustered segments. The intersections of the best fit lines form the building corners. Fig. 7.17 shows an estimated building boundary which is rotated. This portion of the algorithm should be improved in future work.

7.5.3 Complex Building

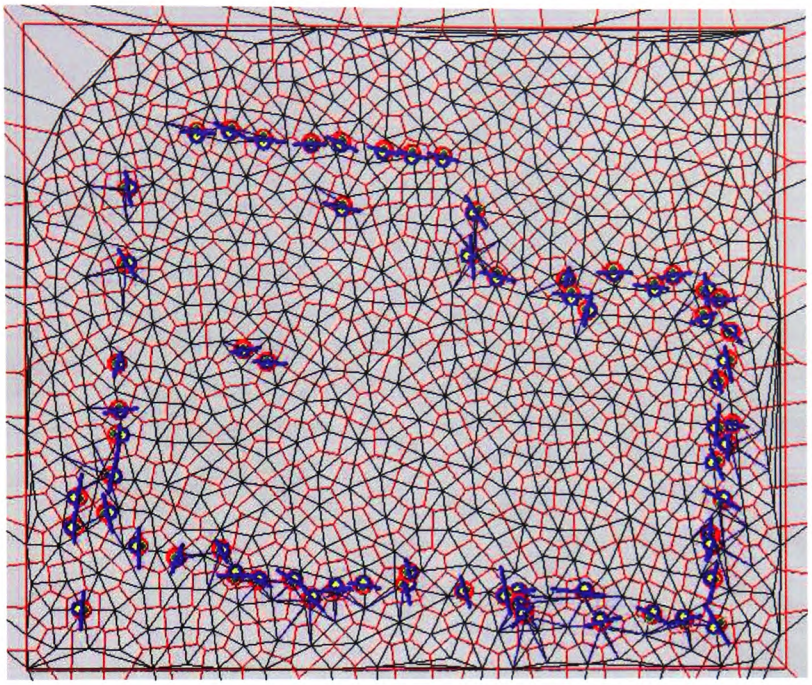
This example is composed of three buildings attached to each other. Fig. 7.18 shows the 2D and 3D views of the example. The boundaries of the whole building are more complicated than usual. A 0.8 metre disc is set to resample the data points to a lower resolution index layer and the vertical wall segments are shown in Fig. 7.19. The clustered vertical wall segments are too scattered to form a closed building outline (shown in Fig. 7.19(b)). With the use of boundary fitting methods, the result may be improved in the future.

7.5.4 Evaluation on Building Boundaries Extraction

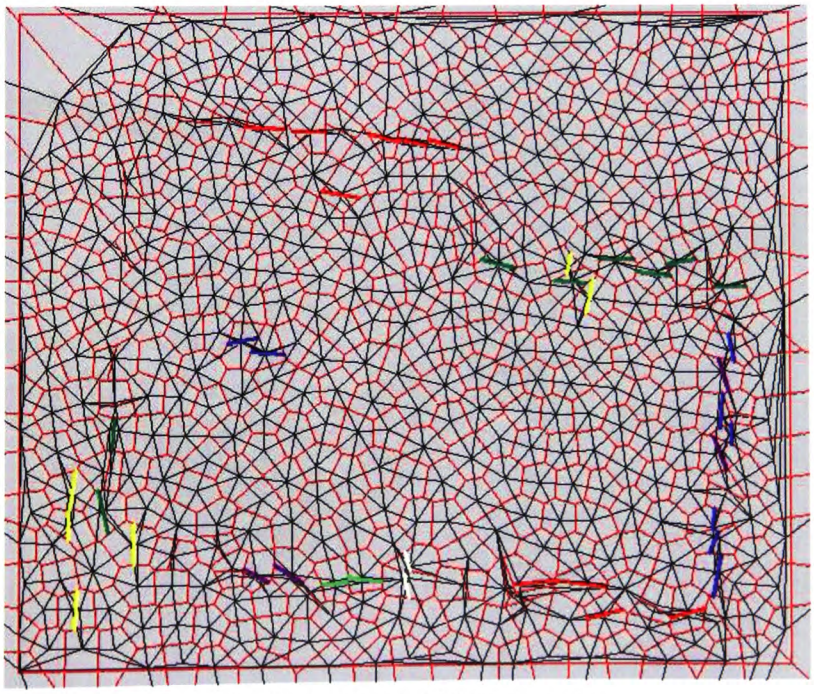
Two L-shaped buildings and a complex building are used to demonstrate the boundaries extraction. There are two main targets to fulfill in the extraction which are to look for the vertical wall segments, and to connect the segments and form the building corners.

The first target is fulfilled when look at the first L-shaped building in Fig. 7.12 and the complex building in Fig. 7.19. Most of the vertical wall segments are successfully found which separate the high and low areas. However more effort has to put on connecting the segments to form the building outline.

The second L-shaped building in Fig. 7.17 shows a better result on searching for the wall segments and connecting them to form building boundaries. Though the building corners are found, further development is needed to improve the quality and accuracy of the result.

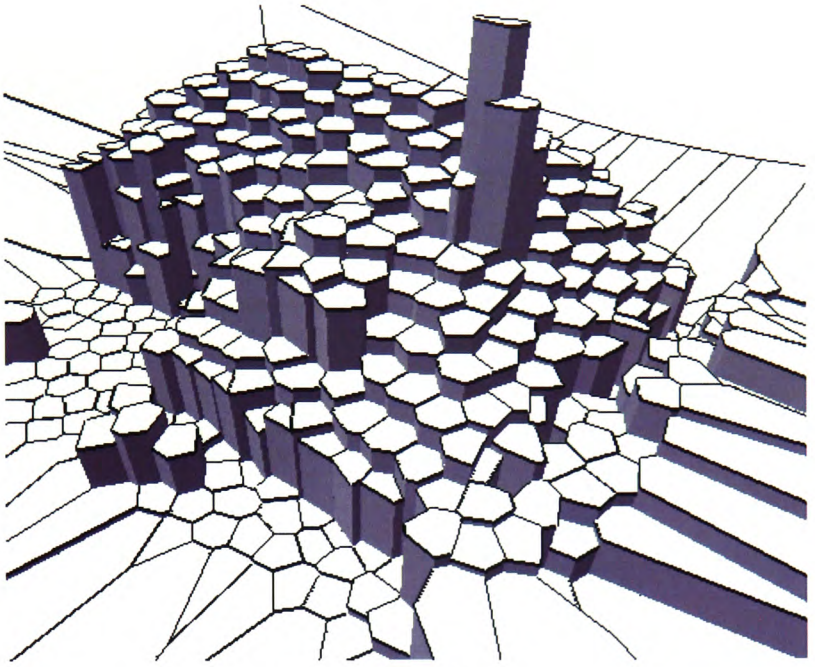


(a) Vertical wall segments

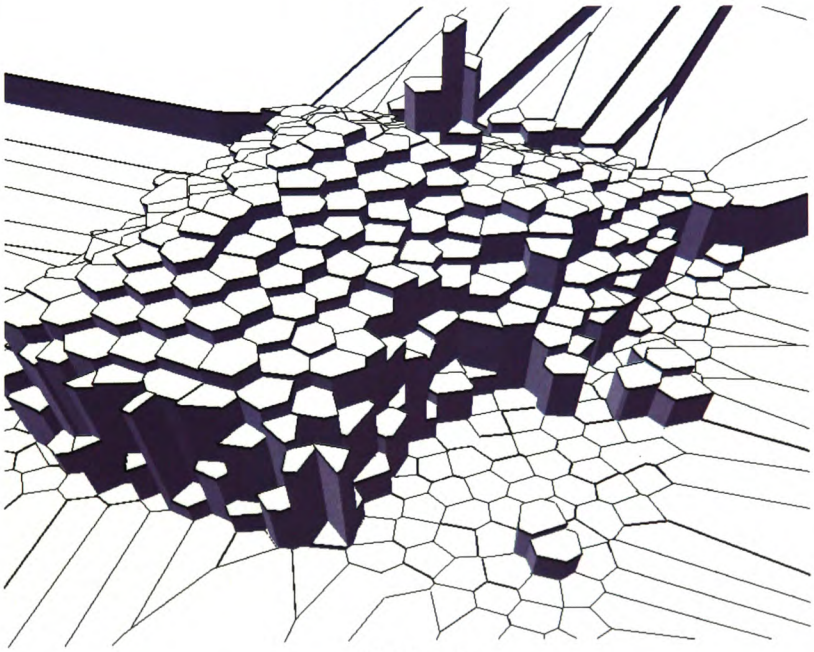


(b) Clustered vertical wall segments

Figure 7.12: The found vertical wall segments

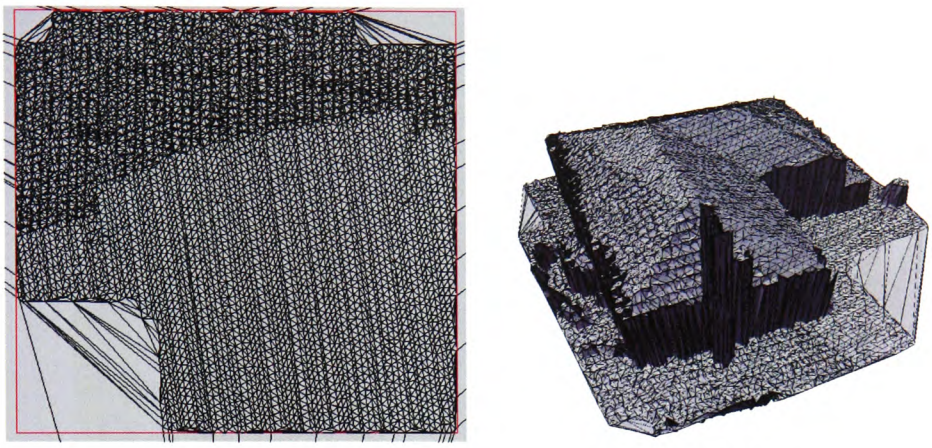


(a) View 1



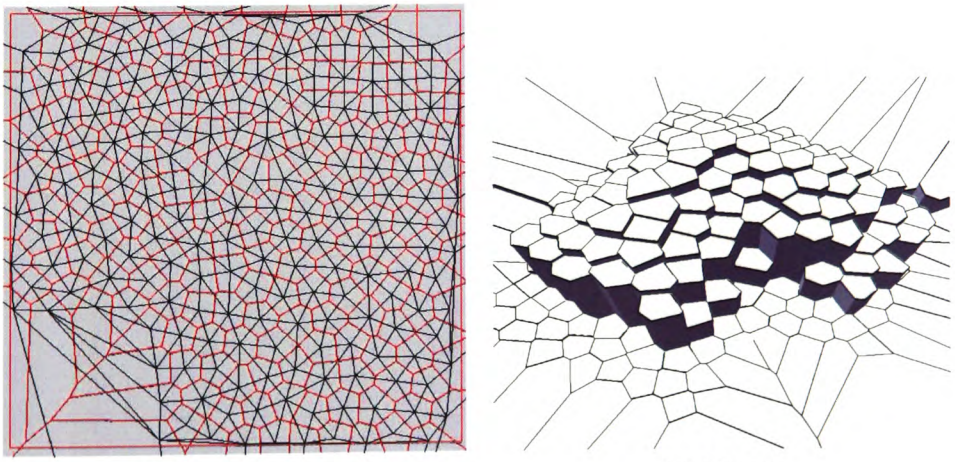
(b) View 2

Figure 7.13: 3D views of the split Voronoi cells



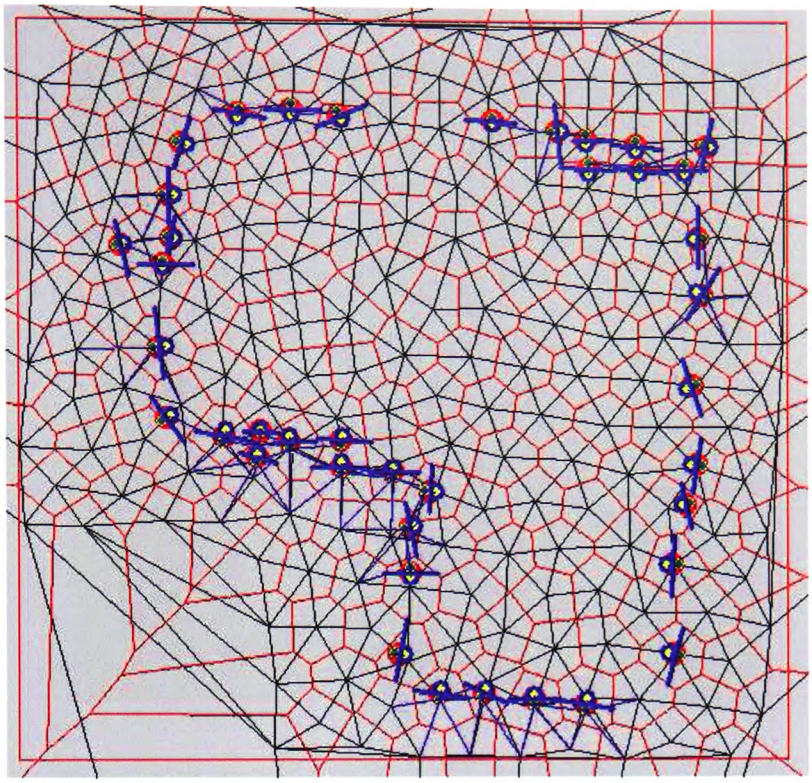
(a) 2D view (b) 3D view

Figure 7.14: An example of L-shaped building 2

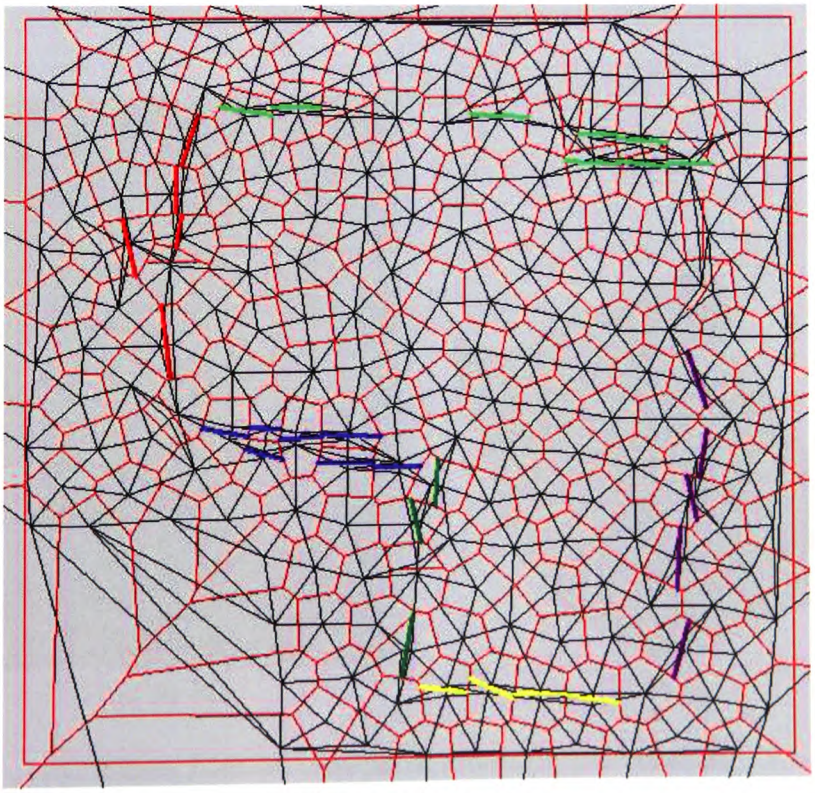


(a) 2D view (b) 3D view

Figure 7.15: The index layer of L-shaped building 2



(a) Wall segments before clustering



(b) Clustered segments

Figure 7.16: The vertical wall segments

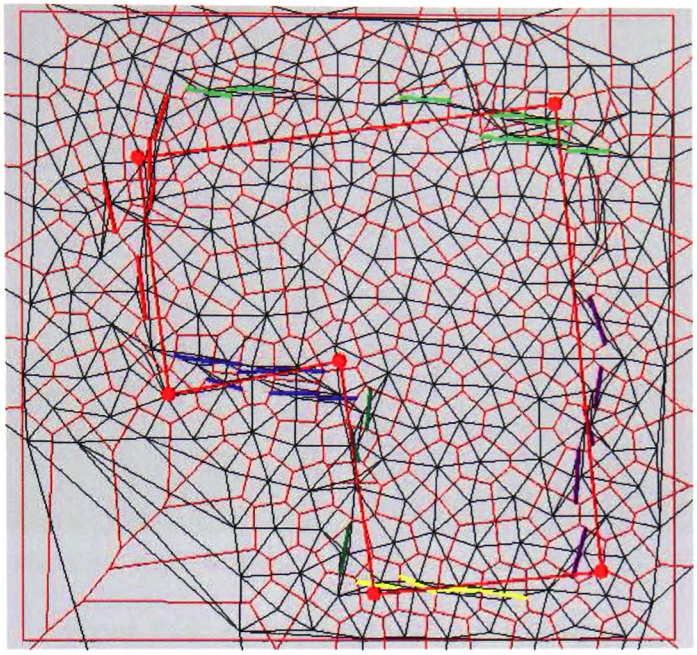
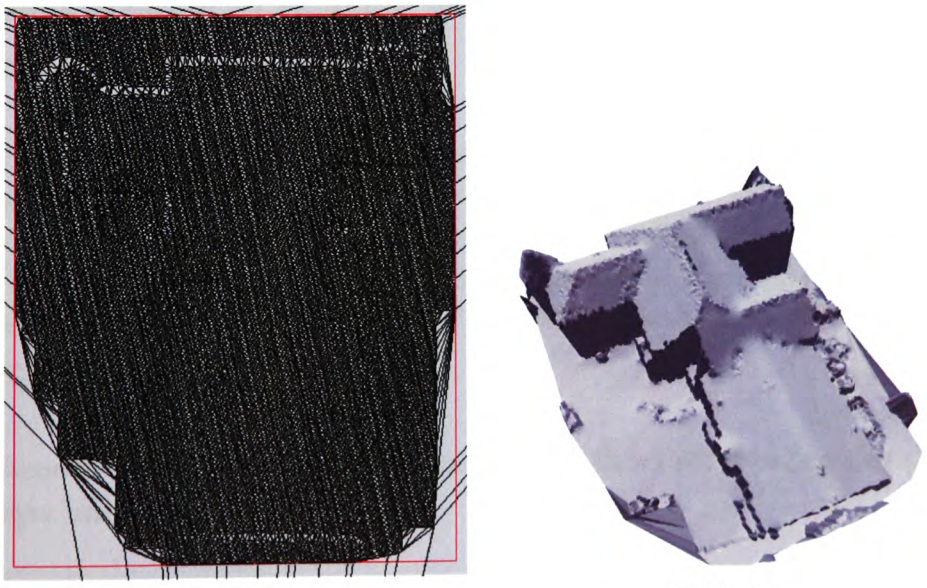
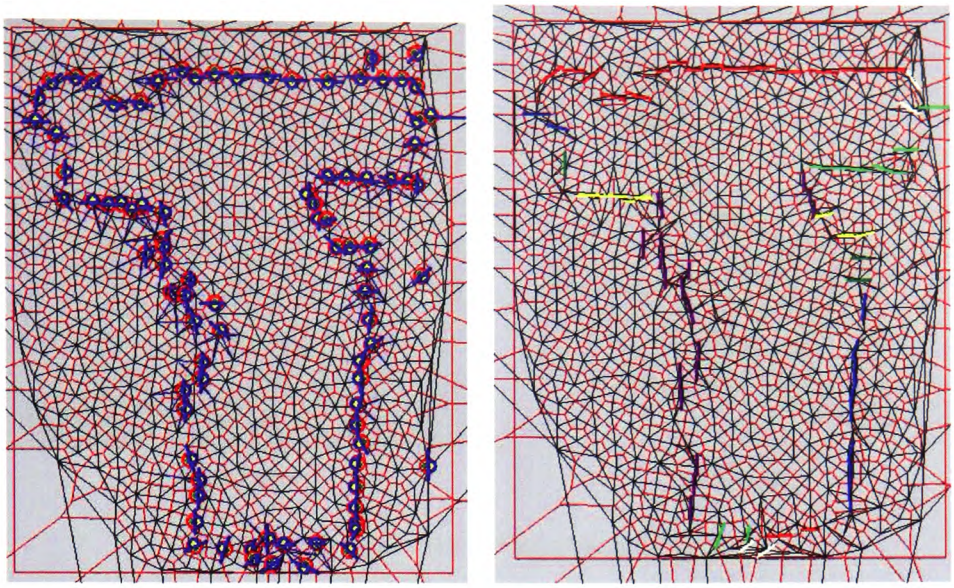


Figure 7.17: 2D View of the estimated building boundaries



(a) 2D view (b) 3D view

Figure 7.18: An example of three attached buildings



(a) The vertical wall segments found (b) The clustered vertical wall segments

Figure 7.19: An example of a complex building

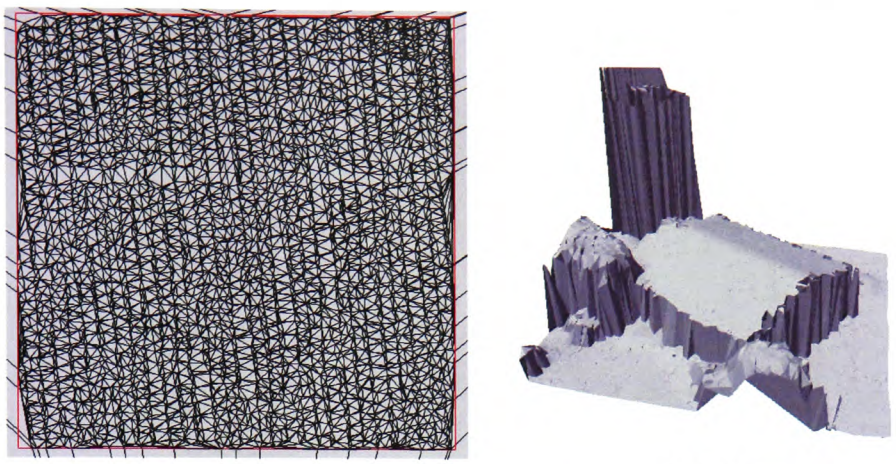
To conclude, further improvement is needed to increase the quality and accuracy on connecting the wall segments and calculating the building corners.

7.6 Demonstration of Building Reconstruction Using Real-World Data

The final test is to reconstruct the whole building including the boundary extraction, roof modelling and building extrusion. Section 7.6.1 shows an example of building reconstruction using the Ordnance Survey Mastermap Topographic Layer and LIDAR data. However there is an offset between the building footprint and the given the O.S. Mastermap Topographic Layer data. Reconstruction using raw LIDAR data in Section 7.6.2 can solve the offset problem by extracting the building footprint.

7.6.1 Reconstruction with the Ordnance Survey (O.S.) Mastermap Topographic Layer and LIDAR Data

An example of a simple building is used to demonstrate the building reconstruction using given the O.S. Mastermap Topographic Layer data and LIDAR data. Fig. 7.20 shows the 2D and 3D views of the building.



(a) 2D view (b) 3D view

Figure 7.20: An example of a simple building

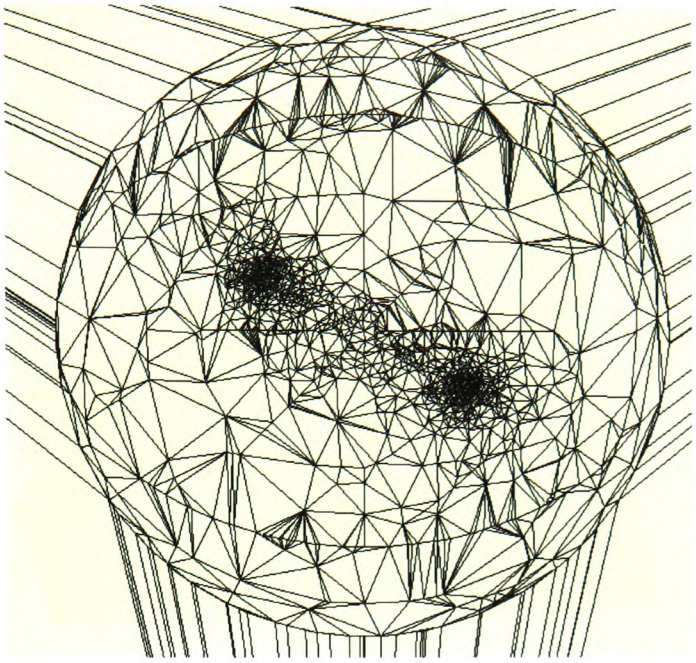
The O.S. Mastermap Topographic Layer data came from an ArcView shapefile. Chapter 6 shows the procedures for reconstructing buildings with the O.S. Mastermap Topographic Layer data. A Delaunay triangulation is created using the raw LIDAR data. Data points inside the given building boundaries are extracted to search for the roof structures in Figs. 7.21 and 7.22. Then the raw LIDAR points inside the building boundaries are deleted. The roof planes intersect the given building boundaries to form corners and roof ridge points. The intersecting points are added on the terrain surface in Fig. 7.23. Fig. 7.24 shows the reconstructed building.

7.6.2 Reconstruction with the raw LIDAR Data

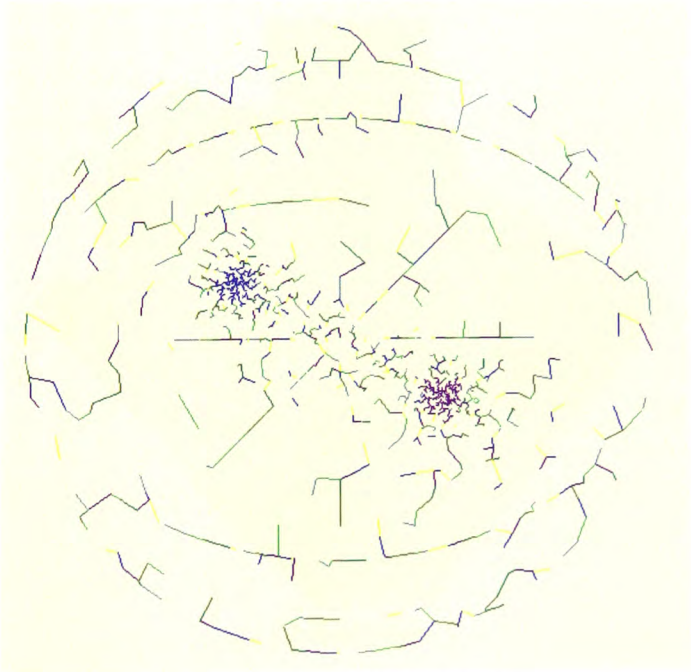
An U-shaped building is used to illustrate the reconstruction. Fig. 7.25 shows the 2D and 3D views of the U-shaped building. The first step is to extract the building block (Figs. 7.26 and 7.27). The next step is to search for the roof structure using the three clustering methods (Figs. 7.26 and 7.27). Then search for the building corners by intersecting the roof planes with the detected building boundaries. The final step is to extrude and reconstruct the whole building.

To find the building block, data points are re-sampled with a disc of 1 metre into a lower resolution index layer in Fig. 7.26(a). Figs. 7.26 and 7.27 shows the procedures for searching and clustering the vertical wall segments, as well as an estimated building outline. Before and after the Voronoi cells are split to search for the wall segments in Fig. 7.28.

Triangles inside the estimated building boundaries are extracted and their normal vectors are plotted onto the unit hemisphere (Fig. 7.29(a)). Triangles are clustered using the three clustering stages (described in chapter 5). Fig. 7.29 and 7.30 shows the 2D and 3D views of the clustered triangles and the roof planes. The unclassified triangles in Fig. 7.30(a) are grouped by the

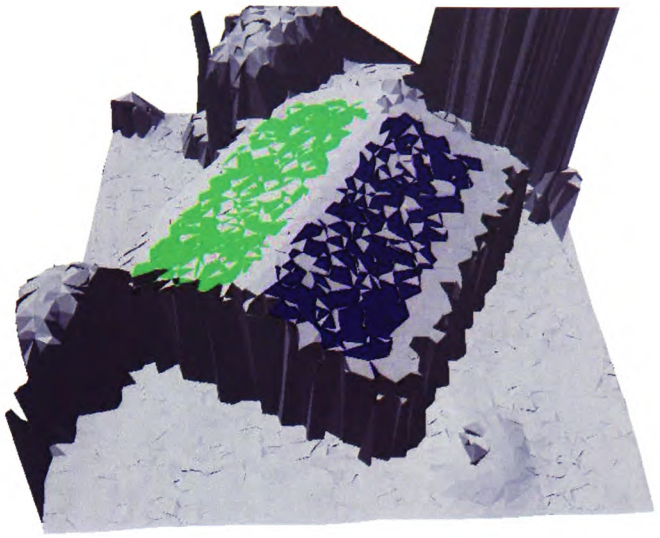


(a) The unit hemisphere

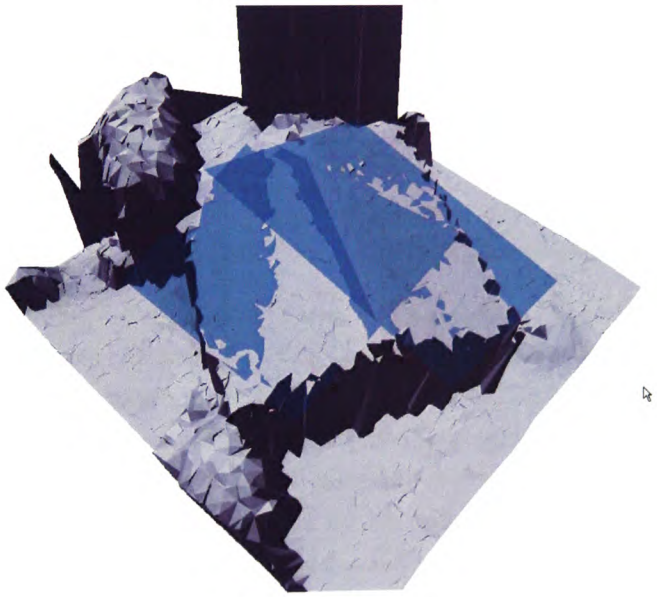


(b) Clustered normal vectors

Figure 7.21: (a)Plotted normal vectors on the unit hemisphere (b)Clustered normal vectors on the unit hemisphere



(a) Clustered triangles



(b) Roof planes

Figure 7.22: (a)The 3D view of the clustered triangles (b)The 3D view of the roof planes

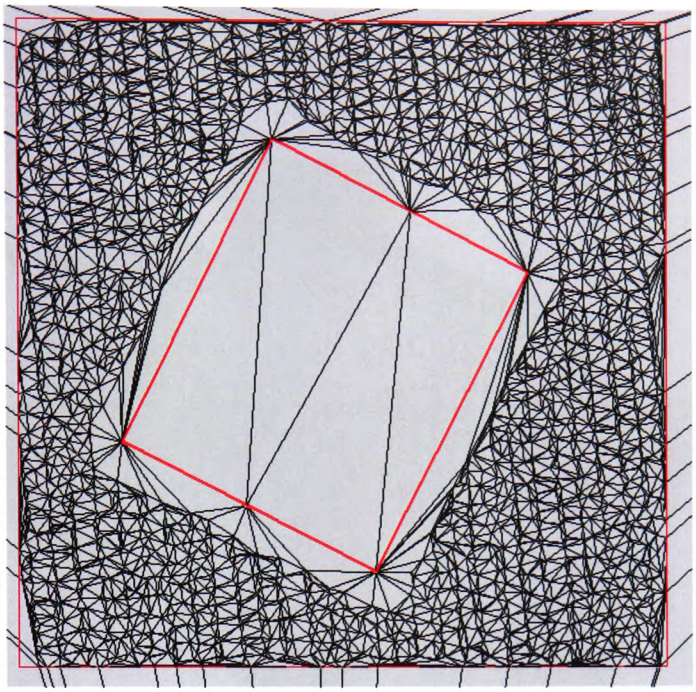


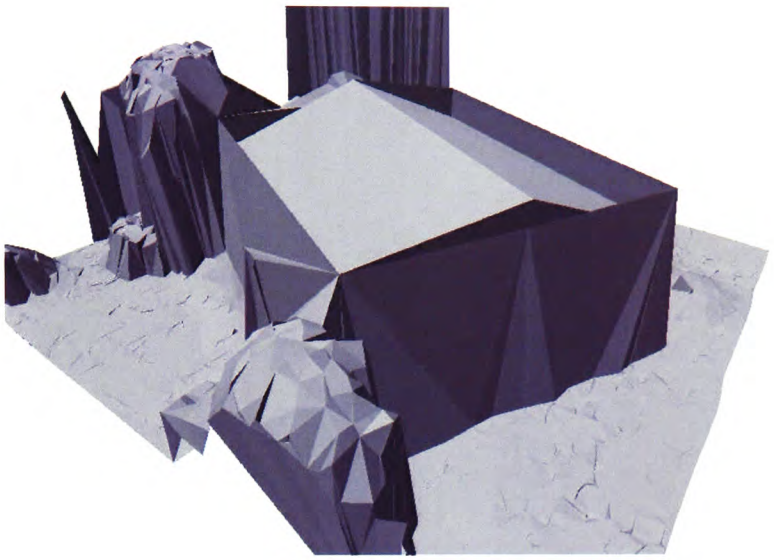
Figure 7.23: 2D View of the estimated building boundaries

region growing method; therefore the relationships between the roof planes are found for the next step (three planes intersection). Fig. 7.30(b) shows the clustered triangles and the created roof planes (transparent blue).

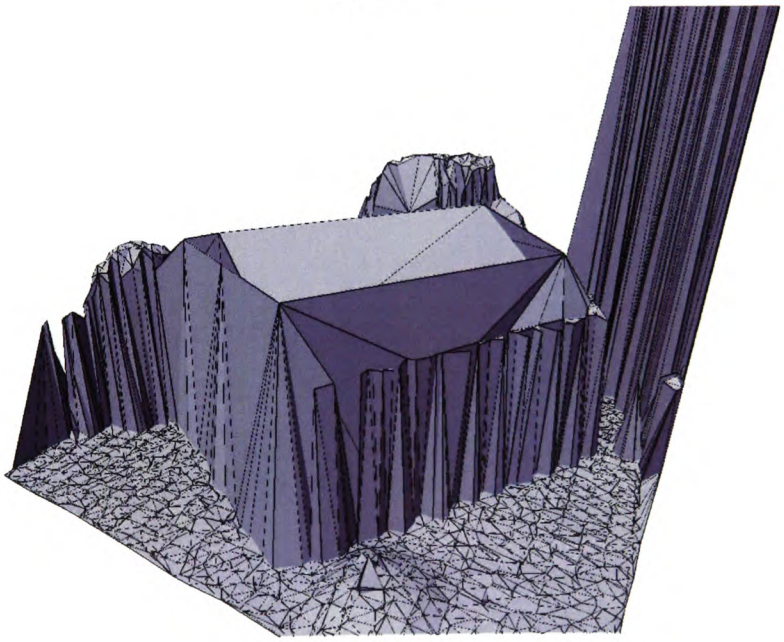
The intersection points are calculated by the three planes intersection and added to the terrain surface in Fig. 7.31. Fig. 7.32 shows a 3D view of the final reconstructed building.

Roof modelling and building boundaries extraction algorithms are the two main methods to be verified in the system. Many different buildings have been used to demonstrate the system in the above sections. The roof planes of most buildings were successfully identified in section 7.3. More problems have been occurred on building boundaries extraction algorithms in section 7.5. Further development for connecting the wall segments and rectangularisation of the building corners are needed. The building outline could not be found; even though the roof planes were found, the building could not be reconstructed successfully.

The objective of the research project is to develop an algorithm which can reconstruct a 3D city model from LIDAR data only with preserved topological connectivity. The above buildings was used to test the reliability of the developed algorithms, and some drawbacks were found and recommendations were made to improve the algorithms. Only several buildings were used to verify the algorithms, and only one building was used to test at each time.

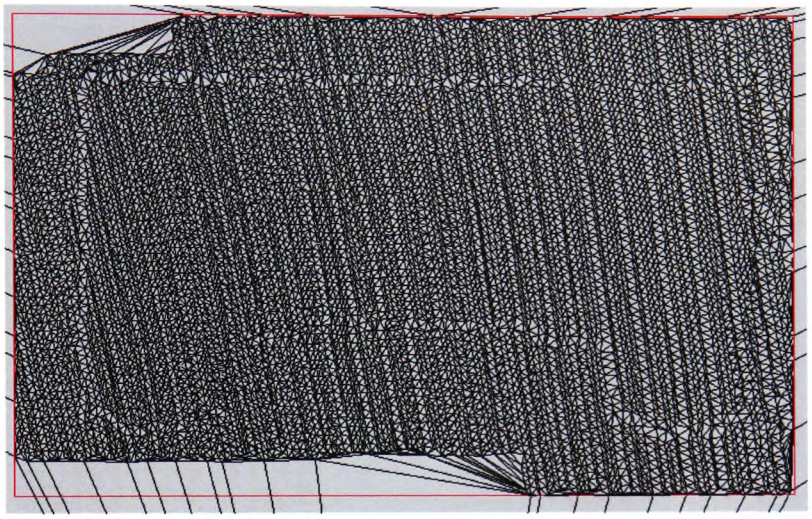


(a) View 1

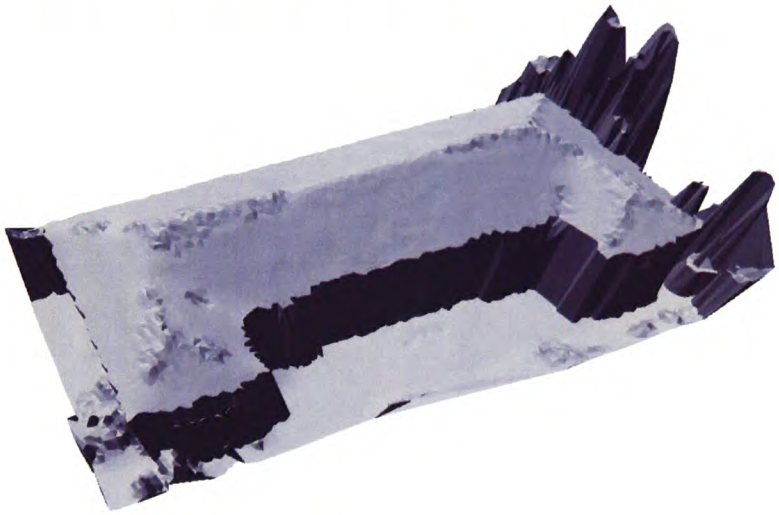


(b) View 2

Figure 7.24: The reconstructed building

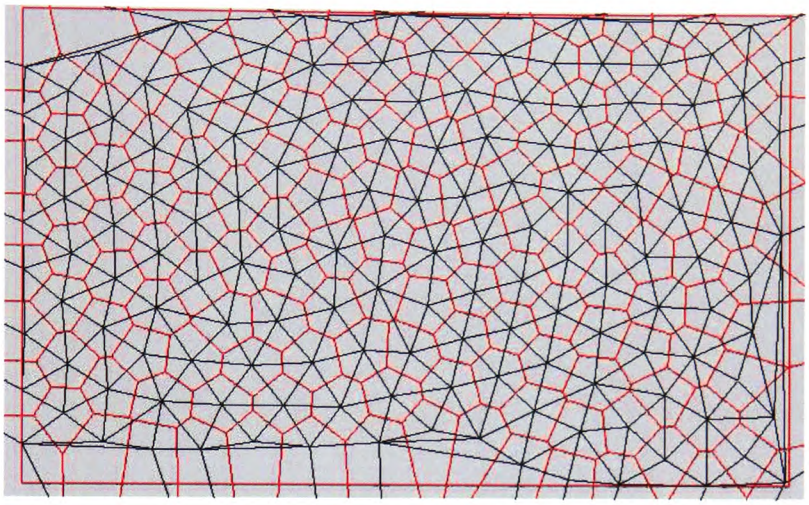


(a) 2D view

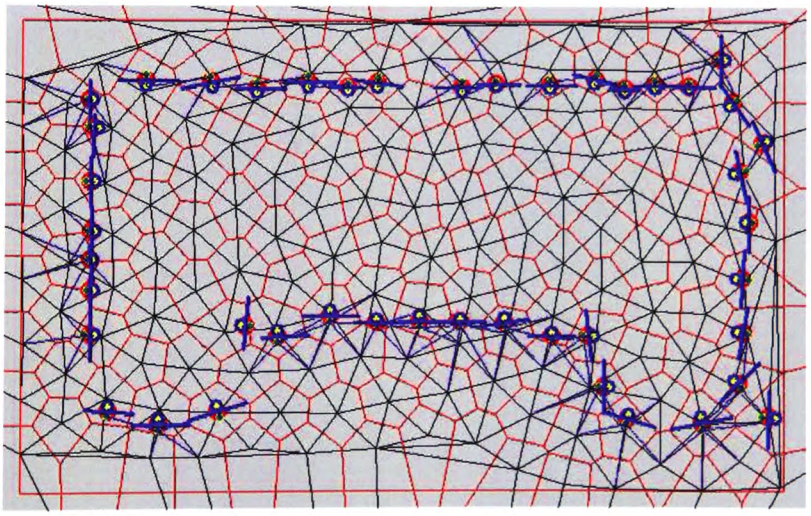


(b) 3D view

Figure 7.25: An U-shaped building

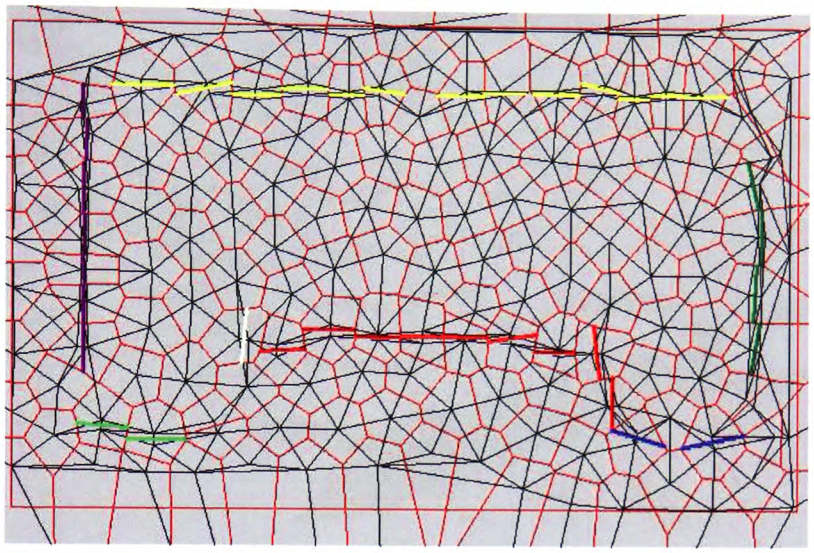


(a) Index layer

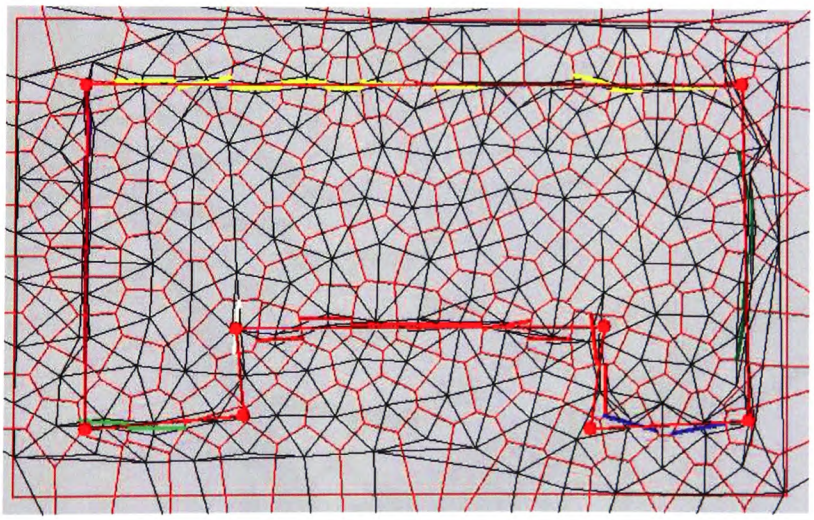


(b) Found segments

Figure 7.26: (a)The resample index layer (b)The vertical wall segments found

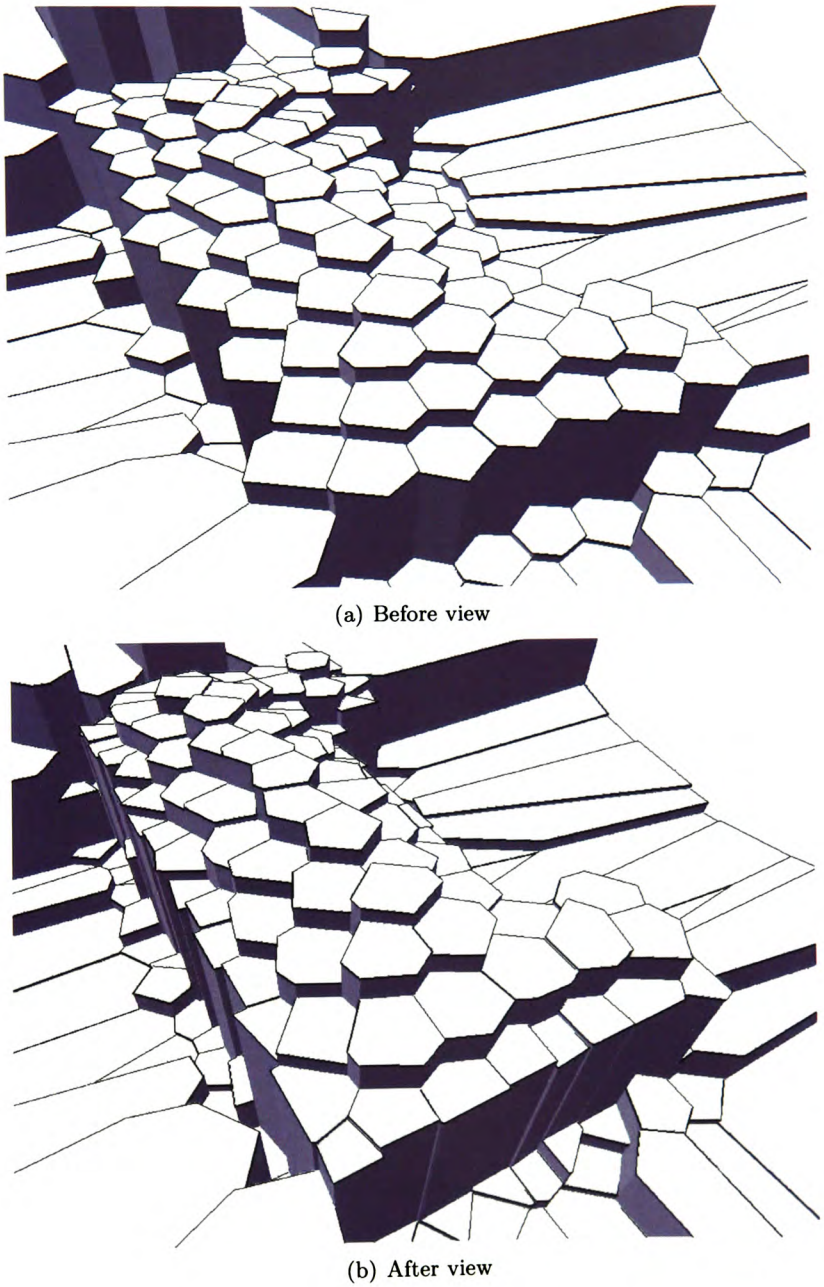


(a) Clustered segments



(b) Intersected points

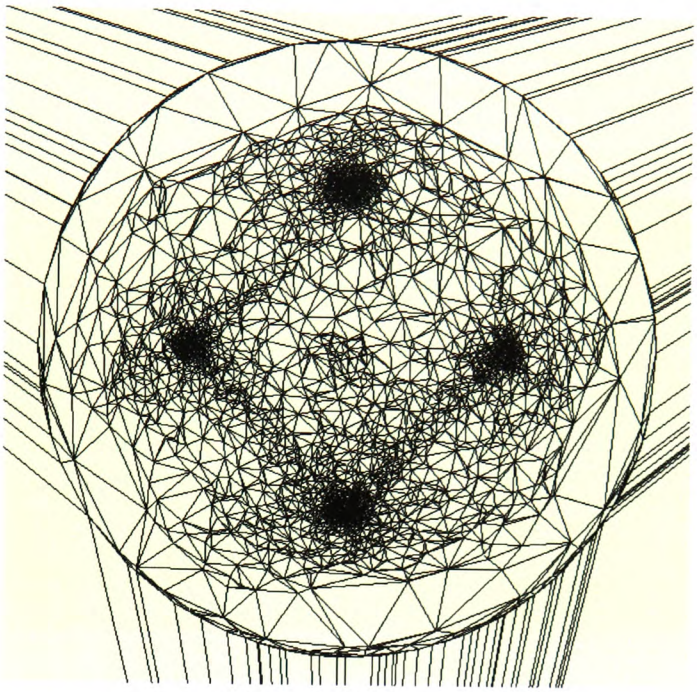
Figure 7.27: (a)The clustered vertical wall segments (b)The estimated building boundaries



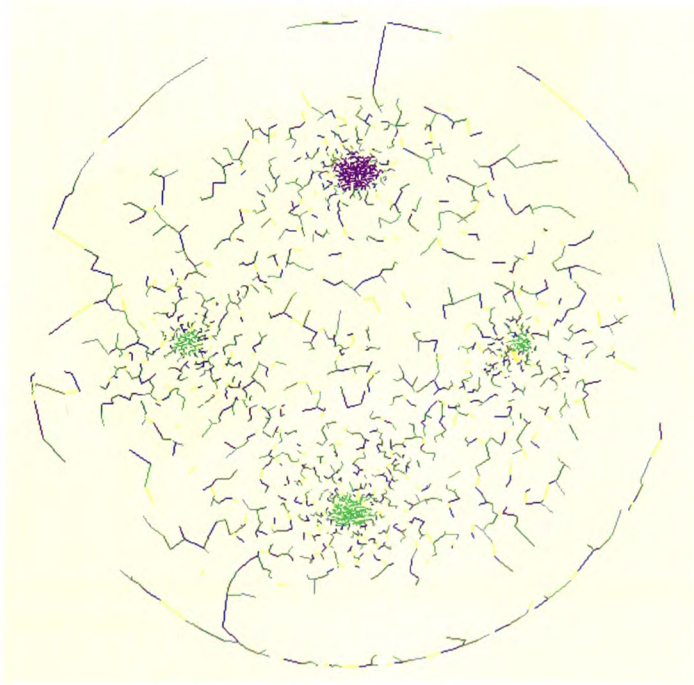
(a) Before view

(b) After view

Figure 7.28: Splitting the Voronoi cells with the wall segments found



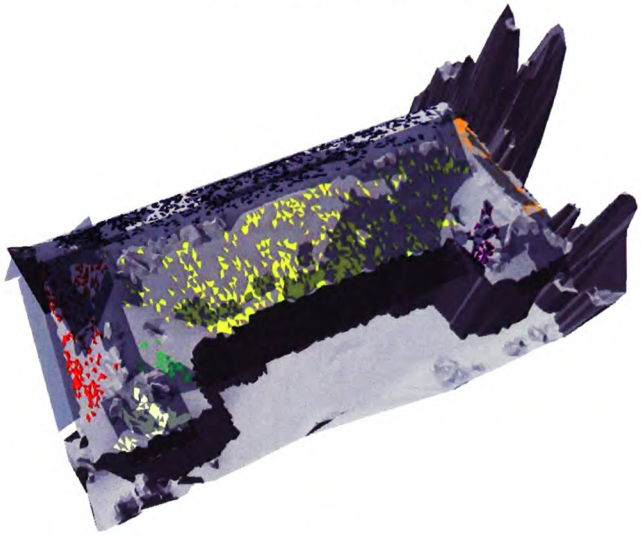
(a) The unit hemisphere



(b) Clustered hemisphere

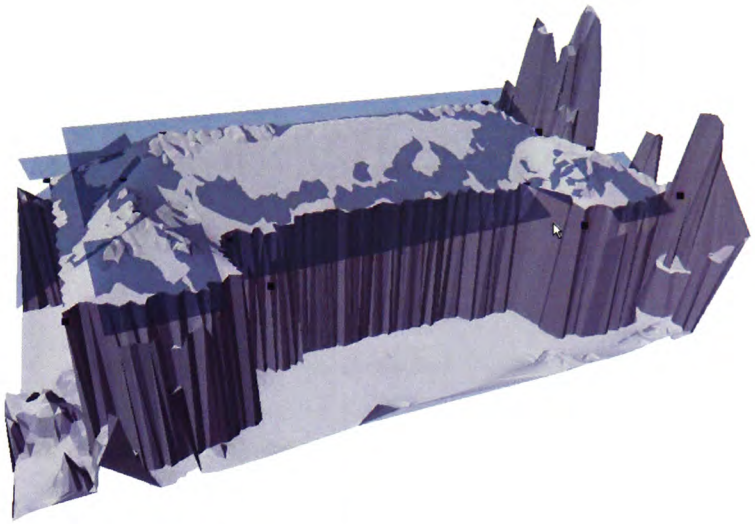
Figure 7.29: (a)The plotted normal vectors on the unit hemisphere (b)The clustered normal vectors

(a) Clustered triangles

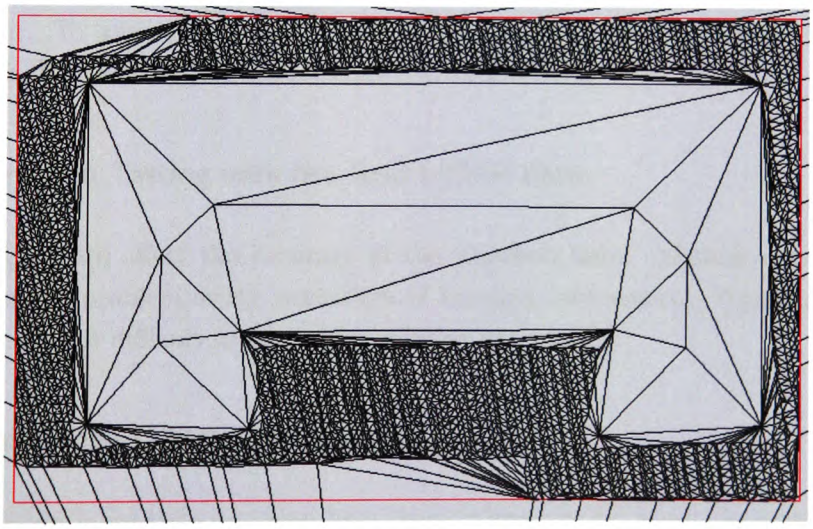


(b) Roof Planes

Figure 7.30: (a)The clustered triangles after region growing (b)The clustered triangles and estimated roof planes



(a) 3D view of the intersection points



(b) The intersection points are added on the terrain

Figure 7.31: Intersection Points

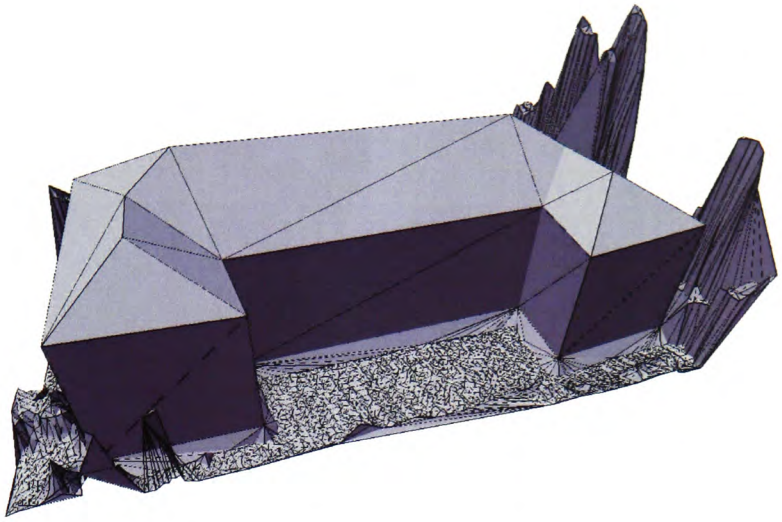


Figure 7.32: The reconstructed U-shaped building

7.7 Evaluation of the Proposed Methods

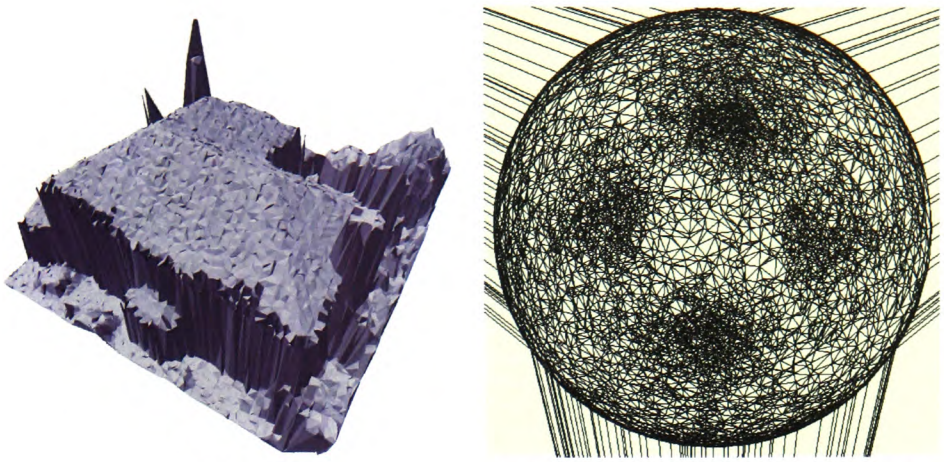
The evaluation of the system are shown in two sections. The first section discusses the problems of using real data. Due to many differences between the real and artificial data, unexpected problems occur. To solve the problem, further development and improvements are needed in future research. General suggestions are shown in section 7.7.2.

7.7.1 Problems on Testing with the Real LIDAR data

Multi-scan flight lines affect the accuracy of the captured data. Missing data points in the shadow area create problem in the extraction of building information. Too much noise in the LIDAR data makes it difficult to locate the building boundaries.

Multi-scan lines of the LIDAR data

If the data contain more than one scan line, there is likely to be some mis-match between the flight lines. Each flight line is affected by the height and altitude of the aircraft as well as drift in the calculation of the GPS or INS system. Even when these are corrected for noise in the data, the triangulated result has normals that the accuracy of the point may decrease. Fig. 7.33(a) shows an L-shaped building which has multi-scanned paths. It is not easy to cluster the normal vectors on the unit hemisphere of the L-Shaped Building (Fig. 7.33(b).



(a) 3D View (b) Normal vectors on the unit hemisphere

Figure 7.33: An L-shaped building

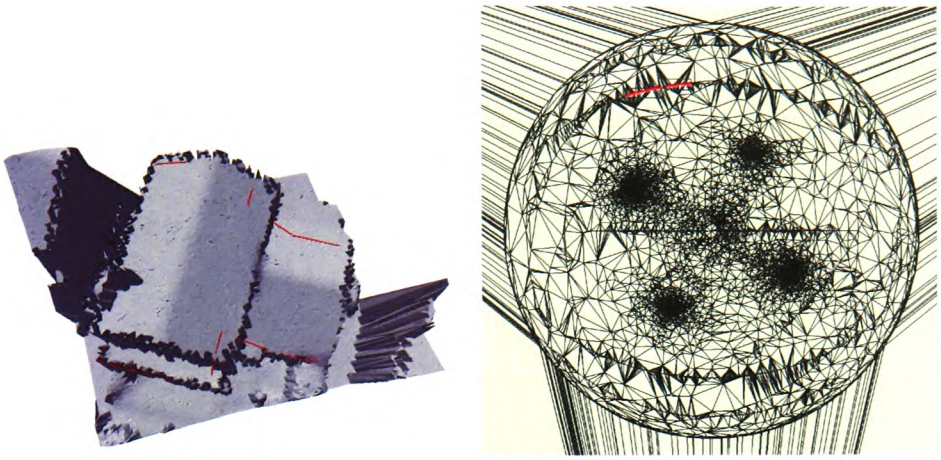
Another example of a T-shaped building is shown in Fig. 7.34(a). When the normal vectors are plotted on the unit hemisphere, some of the normal vectors form a strange pattern (Fig. 7.34(b)). Fig. 7.35 shows some normal vectors on the unit hemisphere to form an arc or a straight line. However the associated triangles in Fig. 7.36 do not show any special pattern. Those triangles are tiny (the blue dots in figures 7.36(a) and 7.36(b)) on the roof and it may be caused by the mis-matching of the multi-scaned data.

Shadow area of the building

Missing data points in shadow areas increase the difficulties in the extraction of building boundaries. Fig. 7.37(a) shows two examples of missing data in shadow areas (the boundaries are drawn by thick red line). The first step in building outline extraction is to resample the data points. It is not possible to use a small disc for making the index layer because too small Voronoi cells in the index layer may contain only a few data points in those areas missing data. To resample the data points in big Voronoi cells may return inaccurate vertical wall segments.

Too Much Noise in the Data

High quality data are used in this project, but real data always contain noise which causes the clustering to have poor or unexpected results. The testing data had been smoothed by the providing company which gave better results.



(a) 3D View (b) Normal vectors on the unit hemisphere

Figure 7.34: A T-shaped building

7.7.2 Improvements for the Proposed System

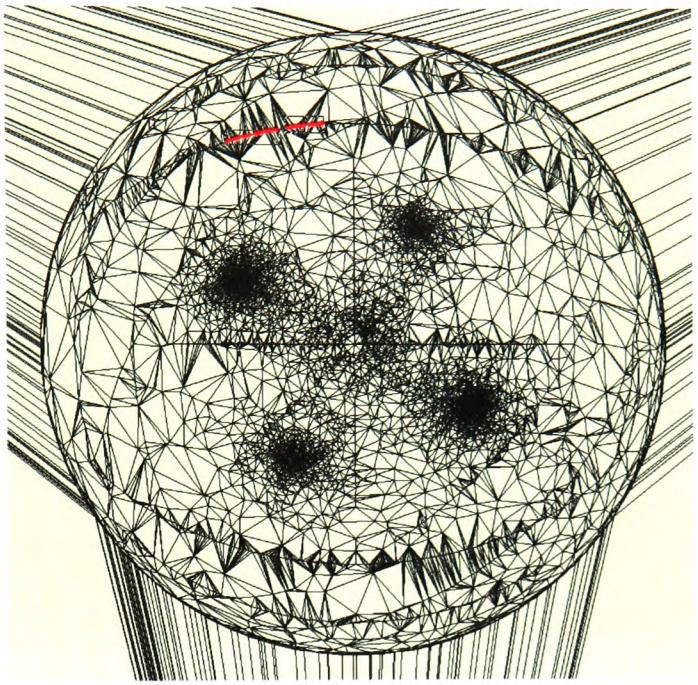
The proposed system has successfully reconstructed the U-shaped building using raw LIDAR data only. However there is still room for improvement in the future. This includes:

1. Consider the way to extract building boundaries when another building exists next to it. It is rare to have a stand alone building; therefore it is necessary to improve the algorithm.
2. Estimate an improved best fit line to describe the clustered split Voronoi edges after finding and clustering the vertical wall segments.
3. Try to avoid using too many tolerances when clustering or searching for objects.
4. Smooth the data locally to improve the clustering result.

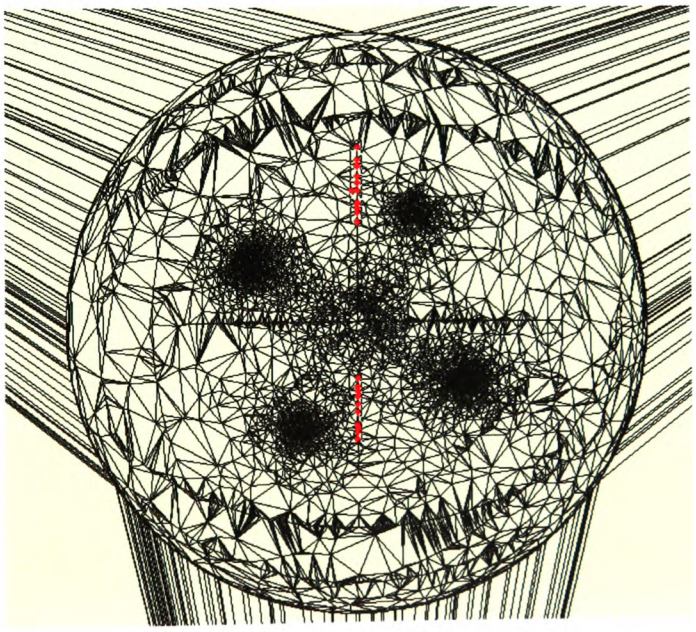
7.8 Chapter Summary

Sections 7.3 and 7.5 demonstrate the roof remodelling and building boundaries extraction methods separately. The algorithms are evaluated individually because it increases the degree of freedom for future applications and modifications.

Evaluation of roof remodelling methods gives a significant result when most of the roof planes are correctly identified by the algorithms. Region growth finds the adjacency relationships between the roof planes. Then another simple Delaunay triangulation and three planes intersection are used to calculate the intersected points between the roof planes and vertical walls. The building corners and roof ridges are correctly formed for building reconstruction. However

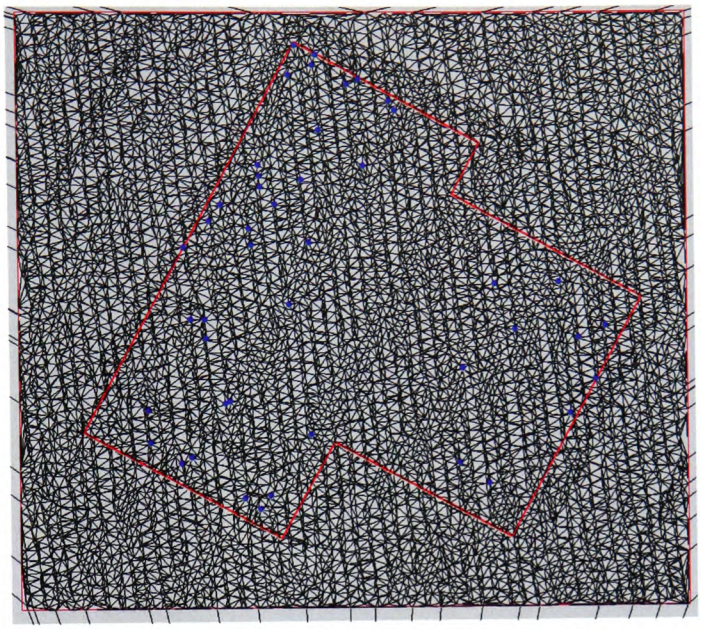


(a) Select plotted normal vectors 1

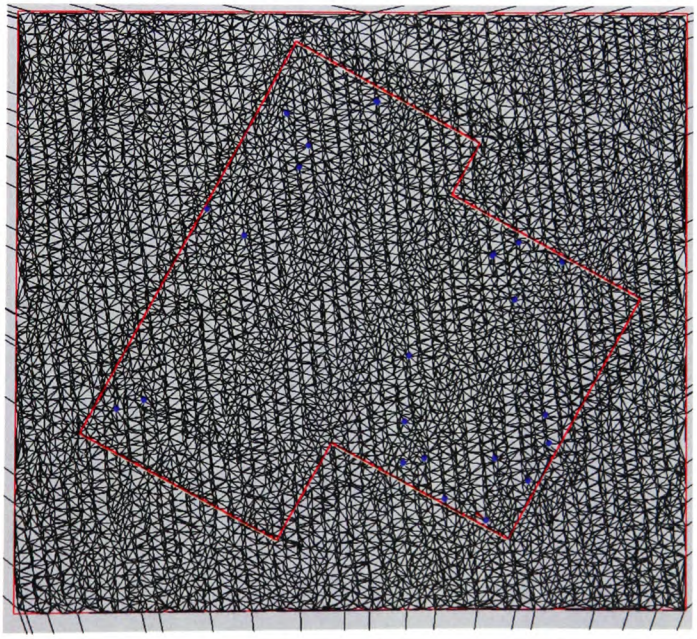


(b) Select plotted normal vectors 2

Figure 7.35: Plotted normal vectors

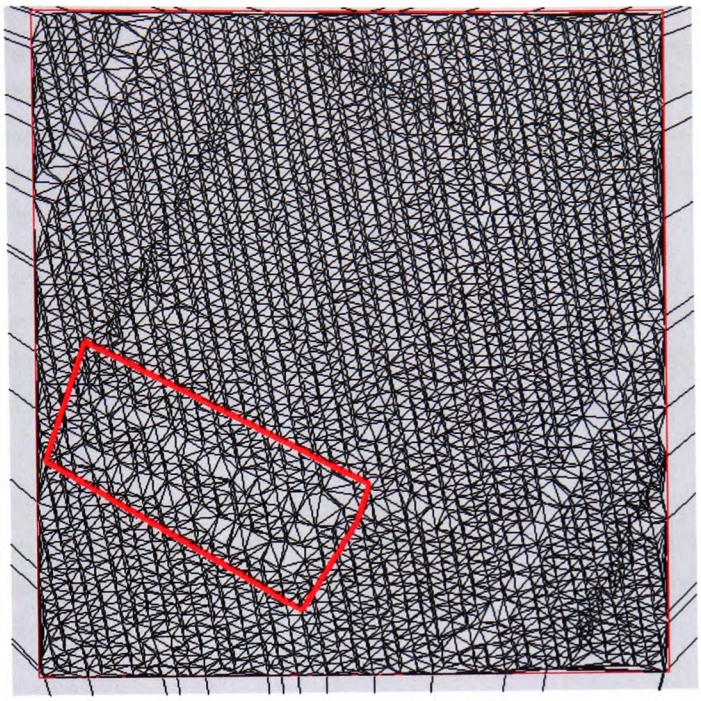


(a) The associated triangles 1 of Fig. 7.35(a)



(b) The associated triangles 2 of Fig. 7.35(b)

Figure 7.36: The associated triangles of Figs. 7.35(a) and 7.35(b)



(a) A T-shaped building

(b) A two plane roofed building

Figure 7.37: Data missing areas

further development, searching for small objects on the roof (for example, dorms and chimneys), will be an essential improvement.

Though the building boundaries extraction method does not perform ideally, it is still an alternative method to search for the building footprint. The idea of looking for vertical wall segments to separate and maximise the height differences, works really well in the above examples. However searching for the connectivity between the wall segments and calculating the averaged lines from the wall segments will be improved by future work.

Section 7.6 demonstrates the system using real data which includes the O.S. Topography Layer and the second pulse LIDAR data. The demonstration uses 2D map and LIDAR data together for the reconstructions which makes an offset between the building and the roof boundaries. When LIDAR only is used to reconstruct the building in Fig. 7.32, this can avoid the offset problem.

There is no perfect solution for automated building reconstruction using LIDAR data only; however the proposed method still can be an alternative solution to the problem. An evaluation of the entire proposed system is made in section 7.7 which reviews the problems and suggests some solutions to the system.

Chapter 8

Conclusions

The use of LIDAR to reconstruct 3D city models has become a popular research topic. However it is not as simple as it looks. The complexity of LIDAR data leads to problems when extracting building information from it. Existing research combines different data sources to extract building information (for example, building outlines, roof structures and so on). However I believe that building reconstruction can be done by using raw LIDAR data only. An alternative approach to extract building boundaries and remodel roof structures is presented in this research.

8.1 Summary of the Research Project

The objective of the research project, defined in Chapter 1, is to reconstruct 3D building models using raw LIDAR data only. The LIDAR data contains a large number of x, y and z coordinates, and does not provide any terrain objects in direct ways. The features can be extracted after segmenting and analysing the raw data. A good extraction algorithm is needed to search for the building information. The algorithm includes searching for the building footprint and the roof style. A good building reconstruction method is required to keep the topological connectivity of the model.

Chapter 2 has reviewed different surveying methods to capture 3D data from the terrain surface. LIDAR looks really suitable to be used in this research.

Much existing research has been done and most use additional data sources, pre-defined building models or high-density data points to reconstruct the buildings. A method is searched which can use the raw LIDAR data to solve the problem. Chapter 3 gives a summary of current research methods.

Chapter 4 shows the first proposed method to extract building boundaries. The building is assumed to be composed of many vertical wall segments which separate the high and low areas. To form the building boundaries, I need to search for and connect those wall segments. Principal

Component Analysis is used to recognise the line of sight which finds the location to split the high and low land. The algorithm is based on the Voronoi diagram.

The next step is to locate the roof structure after the building footprint is found. The shape of the roof is assumed to be made up of different roof planes. Therefore searching for the roof planes is the aim of the second method. Triangles contain implicit information; therefore data points are extracted to create a triangulation. Clustering the triangles into different groups, identify each group of triangles that share the same properties (e.g. orientation and geographical location). A region growing method is used to search for the relationships between the clustered triangles. Each group of triangles represents a piece of a roof plane. After the relationships between these planes are found, three planes intersection is used to calculate the building corners and roof ridges. The procedures of the algorithm are described in the Chapter 5.

The triangulation has been successfully extended with bridges, tunnels and buildings while the topological connectivity is preserved (Tse, 2003). To reconstruct 3D buildings, the first step is to delete raw LIDAR data points which fall inside the building boundaries. Then the building is extruded and the roof is remodelled. Chapter 6 shows the reconstruction procedures.

Synthetic data are used to illustrate the proposed system. Real LIDAR are used to demonstrate the suggested building boundaries extraction and roof remodelling in Chapter 7. This is used to evaluate the methods, to search for potential problems, to suggest any further improvement.

This chapter concludes the research with a project summary, significat and advantages of the proposed system, further development and future application.

8.2 Significant of the Proposed System

The popularity of LIDAR data leads to a need for finding a solution of extracting building information and reconstructing 3D building models. The proposed building reconstruction system is an alternative approach to solve this problem.

Different current research methods have been reviewed and many of them try to fit the best described model to the extracted building information. This method requires a building model library and it takes time to create one. Therefore a method that does not use any pre-defined model is introduced in this project. The proposed approach is to extract information and use the extracted information from LIDAR to recreate the buildings.

One of the advantages of the proposed system is to reconstruct buildings without using any additional data sources. LIDAR data provides the x, y and z coordinates only; therefore building boundaries or roof structures can be obtained by analysing the data. Though additional data sources are handy to use, they may not be suitable for the reconstruction. For example, in Chapter 7 the Ordnance Survey Mastermap Topographic Layer is used to test the system.

However there is a gap between the given building boundaries and the building itself. Most of the building has a roof and normally the roof is bigger than the building footprint. The data from the Ordnance Survey are based on the building footprint; therefore an offset between them occurs. However as LIDAR is captured from the top of the building, extract building outlines from LIDAR can solve the problem.

Rottensteiner (2001); Khoshelham (2005); Sohn and Dowman (2001) tried to extract building boundaries from aerial images and remodel the roof structure using LIDAR data. To extract building boundaries from a pair of aerial images, it used to take a long time. The proposed system can be automatic without any human interaction. The procedures from loading the raw LIDAR data until displaying the reconstructed building models can be run automatically. Minimising the degree of human input reduces the cost of building reconstructions.

Many of the current roof remodelling algorithm has similar approach to this project which requires the computation of surface normal vectors (Elaksher and Bethel, 2002a; Brenner and Haala, 1998; Hofmann et al., 2003; Hofmann, 2004). Some of them were using a region-growing method to reconstruct the roof planes, if the vectors tend to be very noisy that some of the roof planes are disconnected. It may not be possible to find the connectivity between planes. Hofmann (2004) found the relationship between the roof planes according to their overlapping areas. It may not be accurate to use the roof size or overlapping area at this stage because the roof planes were not properly formed at the moment. According to our approach, all the clustered triangles formed infinite planes and a separate Delaunay Triangulation was created to find the relationship between the roof planes, which could prevent any inaccurate guessing. The use of the normal vectors approach seems really attractive because of easy understanding and low calculation time; however many threshold has been set during the processing time. Various of laser data points (for example, density and quality) can be used to do the statistical analysis for fixing the threshold in the future development.

The 3D Hough transform Vosselman (1999a,b); Vosselman and Dijkman (2001) was used to extract roof planes which gave a really good result. However the algorithm requires much higher point density and complicated calculation process. If normal vectors can be computed accurately which can be used in the Hough transformation, it can speed up the processing time and increase the reliability. With the advanced technology, the quality of the laser data points can be improved easily. Therefore the combination of the 3D Hough transform using normal vectors algorithm can serve a better solution for the roof remodelling.

Some research has developed approaches for roof modelling which are complicated for further application. The proposed method of this project is straight forward and easy to understand, and can be easily applied to different systems. In roof remodelling, region growing is used to find the adjacency between the roof planes. Region growing is not used to search for triangles and roof planes which can avoid losing data.

A three planes intersection is used to calculate the intersected points between roof planes and vertical walls. A simple Delaunay triangulation is created to recognise the connections between the intersected points. The simple Delaunay triangulation works easily to solve the problem.

Among the current building reconstruction algorithms from LIDAR, no one can recreate the 3D buildings with preserved topological connectivity. The use of Euler Operators to reconstruct the 3D buildings allows the user to modify and manage the terrain model easily. Topological connectivity is preserved for future spatial analysis.

The whole system is based on Delaunay triangulation and Voronoi diagrams. The advantages of using them are:

- more flexible when splitting the cell. The Voronoi cell can be split in different shapes, which is really useful.
- more representative when re-sampling data points. Though different re-sampling methods are used in this research, this can be a suggestion for further development. Smaller sampling cells (more sampling points) can be used to represent an area which has a higher density of data points.

The system is flexible because different part of the system can be replaced or modified using other methods. For example, users can use existing two dimensional building outlines as the building boundaries, or other roof-remodelling and reconstruction tools can be used to create the building model.

This research is under one single development platform which keeps the environment unique and easily modified. Because of not involving any commercial packages, it can save some production costs.

8.3 Future Development

The system may not work perfectly with all real data; however to a certain extent it shows the possibility and capability of the idea. Building boundary extraction and roof remodelling are the two algorithms developed in the project. Several things can be done to improve the two algorithms. The following two sections suggest some improvements for future development.

8.3.1 Suggested Improvements for Building Boundary Extraction

Two building boundaries extraction algorithms have been developed. The statistical method maximises the height differences between high and low areas, and then splits the Voronoi cells to get vertical wall segments. The building outline is formed by connecting the vertical wall

segments. The method does not give a perfect result; therefore further development is needed to improve the method.

Before finding the vertical wall segments, origin data points are extracted to resample into a lower resolution as an index layer. The current method sets a disc to form the index layer. However it can fix a number of points inside each cell to create the index layer. Small Voronoi cells are used to represent an area with a high density of data points. The index layer can be created by the new method which makes it more representative.

The method of connecting the vertical wall segments does not work well; therefore further development must focus on connecting methodology. Most of the wall segments are found and clustered successfully according to their orientation. Due to the noisy data, the split Voronoi edges cannot be connected easily. Some gaps appear between the clustered wall segments. If further development allows estimating of the adjacent wall segments, it may improve the results.

Each group of wall segments is used to form an averaged line. The current method is to find a best fit line by the orthogonal regression; and the line is calculated using the vectors of all the wall segments. The starting, ending and middle points of each wall segment can be used to calculate the best fit line in the future; which may improve the quality of the results. Normally building corners are rectangular (90 degrees); however it is not easy to get the found building corners to be 90 degree angles. A method to check and rectify the building corners would be significant.

Since 3D Hough transform and RANSAC algorithms (Tarsha-Kurdi et al., 2007; Vosselman, 2003) give a relatively good and accurate result on building boundary extraction. The system can be improved by combining the two algorithms to extract building footprint.

The solution to the missing data problem is described in Chapter 7. Subject to the flight direction, some data may be captured in the building shadows. It works better if the system can estimate the vertical wall segments using nearby wall segments.

The Voronoi City Model method successfully moves the Voronoi cells according to height; however it is difficult to use the boundary of the Voronoi cells to form a straight building outline. Further development is needed to obtain rectangular building boundaries, which may be improved by splitting the Voronoi cells.

8.3.2 Suggested Improvements for Roof Remodelling Method

Data points inside the building boundaries are extracted to form a triangulation. Three clustering methods are used to separate the triangles. Each group of triangles represents a roof plane.

Intersections between the roof planes and vertical walls find the building corners and points of the roof ridges.

The clustering methods work very well to separate the triangles according to their orientation and geographical location. Several tolerances need to be set in each clustering. On the first orientation clustering, the user has to input a tolerance and minimum number of triangles to run the program which decreases the degree of automation. The matter of setting the tolerance can be solved by finding an averaged number. For example, calculate mean, variance or the normal distribution between points.

The clustering methods may be used to search the vertical walls of buildings in the future. Normally building walls are vertical standing along each side of the building. If the normal vectors of the vertical walls are projected onto the unit hemisphere, they should be on or near the edge of the hemisphere. It will be easy to extract them and use the other two clustering methods to find all the vertical walls. However further development is needed to make the idea work.

8.4 Further Recommendation on the Proposed System

The basic ideal of the proposed system focuses on reconstructing building models from LIDAR data. Further development can be done to complete the system.

At present the developed algorithm does not consider the storage of the reconstructed city model. The reconstructed city model will be possible to store if the building information can be recorded in a file. It is a waste of time to run the system and reconstruct the building on the fly every time. A database can be setup to store the information of the terrain model and the building (for example, building corners and the height of the building). The recreated 3D buildings can be stored or kept in the database. Further algorithms have to be developed to split the 3D building models from the terrain to store as a separate object in the database. It can be an interesting improvement for the system.

In the future, there is a possibility to combine a new and innovation concept to reconstruct the 3D city model, which is the City Geography Markup Language (CityGML). It is a an open data source which reach an international standard for spatial data exchange (Open Geospatial Consortium). Targeted developed system will allow generalisation between thematic classes, aggregations, relations between objects and spatial properties.

This research gives an alternative approach to solve the problem of building reconstruction using raw LIDAR data. The above recommendations may improve the system in the future. The quality of LIDAR data is becoming better and better. The Environment agency captures

data with spatial resolutions down to 25cm. The huge improvements in data quality will help with further analysis and give desirable results.

Bibliography

F. Ackermann. Airborne laser scanning - present status and future expectations. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(1):64–67, 1999.

P. Axelsson. DEM generation from laser scanner data using adaptive TIN models. *In IAPRS*, 33(Part B4/1):110–117, 2000.

P. Axelsson. Processing of laser scanner data - algorithms and applications. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54:138–147, 1999.

B. Babu Madhavan, C. Wang, H. Tanahashi, H. Hirayu, Y. Niwa, K. Yamamoto, K. Tachibana, and T. Sasagawa. A computer vision based approach for 3D building modelling of airborne laser scanner DSM data. *Computers, environment and urban systems*, 30:54–77, 2006.

C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *Proceeding of CVPR*, 1999.

C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. *IAP*, 32 Part 3-2W5, 1999.

E. P. Baltsavias. A comparison between photogrammetry and laser scanning. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2):83–94, 1999a.

E. P. Baltsavias. Airborne laser scanning: existing systems and firms and other resources. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54:164–198, 1999b.

B. G. Baumgart. Winged edge polyhedron representation. Technical report, Stanford University Computer Science Department, Stanford Artificial Intelligence Report No. CS-320, 1972.

R. Bimal and S. Kumar. A new approach to polygonal approximation. In T. Ho and G. S. di Baja, editors, *Pattern Recognition Letters 12*, pages 229–234. International Association for Pattern Recognition, 1991.

R. Bimal and S. Kumar. An algorithm for polygonal approximation of digitized curves. In G. S. d. B. T.K. Ho, editor, *Pattern Recognition Letters 13*, pages 489–496. International Association for Pattern Recognition, 1992.

W. F. Bonn. 3D-city models: Automatic and semiautomatic acquisition methods. In D. Fritsch and R. Spiller, editors, *Photogrammetric Week '99'*, pages 291–303, Wichmann Verlag, Heidelberg, 1999.

P. Bourke. The intersection of three planes, October 2001. URL `http://local.wasp.uwa.edu.au/~pbourke/geometry/3planes/`. Equation of three planes intersection.

I. Braid, R. Hillyard, and I. Stroud. Stepwise construction of polyhedra in geometric modelling. In K. Brodlie, editor, *Mathematical Methods in Computer Graphics and Design*, pages 123–141. A Subsidiary of Harcourt Brace Jovanovich, Leicester, 1978.

C. Brenner. Towards fully automatic generation of city models. *Int. Arch. Photogramm. Remote Sensing*, 32 Part 3.:85–92, 2000.

C. Brenner and N. Haala. Rapid acquisition of virtual reality city models from multiple data sources. *Int. Arch. Photogrammetry Remote Sensing*, 32 Part 5:H. Chikatsu and E. Shimizu, 1998.

F. Bretar, M. Chesnier, M. Rous, and M. Pierrot-Deseilligny. Terrain modeling and airborne laser data classification using multiple pass filtering. In *Geo-Imageery Bridding Continents, XXth ISPRS Congress, Commission 3*, Istanbul, Turkey, 2004.

R. Brinkman and C. O'Neill. LIDAR and Photogrammetric mapping. In *The military engineer, no. 605*, 2000.

M. A. Brovelli, M. Cannata, and U. M. Longoni. Managing and processing LIDAR data within GRASS. In *The GRASS Users Conference 2002*, Trento, 2002.

A. Brunn. Statistical interpretation of DEM and image data for building extraction. *Ascona01*, pages 171–180, 2001.

A. Brunn and U. Weidner. Extracting buildings from digital surface models. In *3D Reconstruction and Modeling of Topographic Objects*, volume 32, Part 3-4W2, Stuttgart, 1997a. IAPRS.

A. Brunn and U. Weidner. Discriminating building and vegetation areas within digital surface models. Technical report, Institute fur Photogrammetrie, Bonn, 1997b.

R. Burtch. LIDAR principles and applications. In *2002 IMAGIN conference, Traverse City, MI*, 2002.

H. A. K. Charlesworth, C. W. Langenberg, and J. Ramsden. Determining axes, axial places and sections of macroscopic folds using computer-based methods. *Candian Journal Earth Science*, 13:54–65, 1975.

L. P. Chew. Constrained Delaunay triangluations. In *Proceedings of the third annual symposium on computational geometry*, pages 215–222. Annual Symposium on Ccomputational Geometry, Waterloo, Ontario, Canada, 1987.

M. Dakowicz and C. M. Gold. Extracting meaningful slopes from terrain contours. In P. Sloot, C. Tan, J. Dongarra, and A. Hoekstra, editors, *In Proceedings : Computational Science - ICCS 2002, Lecture Notes in Computer Science*, volume 2331, pages 144–153, Amsterdam, The Netherlands, 2002. Springer-Verlag Berlin Heidelberg.

P. H. Dana. Global positioining system overview, 2000.

B. N. Delaunay. Sur la sphere vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.

O. Devillers. On deletion in Delaunay triangulations. In *Symposium on Computational Geometry*, pages 181–188, 1999.

Y. Dobashi, T. Haga, H. Johan, and T. Nishita. A method for creating mosaic images using voronoi diagrams. In I. Alvaro and P. Slusallek, editors, *Proceedings of Eurographics 2002*, volume I, pages 341–348, 2002.

A. F. Elaksher and J. S. Bethel. Reconstructing 3D buildings from LIDAR data. In *Photogrammetric Computer Visionm, ISPRS Commission III, Symposium 2002*, volume Part A, pages 91–96, Graz, Austria, 2002a.

A. F. Elaksher and J. S. Bethel. Building extraction using LIDAR data. In *ACMS-ASPRS 2002 annual conference proceedings*, volume CD-ROM, page 8 pages, 2002b.

M. Elmqvist. Gorund estimation of laser RADAR data using active shape models. In K. Torlegard, editor, *OEEPE workshop on airborne laserscanning and interferometric SAR for detailed digital elevation models*, page paper 5 (8 pages), Royal Institude of Technology, Department of Geodesy and Photogrammetry, 100 Stockholm, Sweden, 2001.

M. Elmqvist, E. Jungert, F. Lantx, A. Persson, and U. Soderman. Terrain modelling and analysis using laser scanner data. *IAPRS*, XXXIV - 3/W4 Annapolis, MD:219–227, 2001.

A. Fischer, T. H. Kolbe, F. Lang, A. B. Cremers, W. Förstner, L. Plumer, and V. Steinhage. Extracting buildings from aerial images uisng hierarchical aggregation. *CVIU*, 72(2):195–203, 1998.

M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6): 381–395, 1981.

M. Flood. LIDAR activities and research priorities in the commercial sector. In *IAPRS WG IV/3*, volume XXXIV, pages 678–684, Annapolis, MD, 2001.

G. Forlani, C. Nardinocchi, M. Scaioni, and P. Zingaretti. Building reconstruction and visualization from LIDAR data. In *ISPRS International Workshop WG V/4 & INTCOM III/V, Vision Techniques for digital architectural and archaelogoical archives*, pages 151–156, Ancona, Italy, 2003.

G. Forlani, C. Nardinocchi, M. Scaioni, and P. Zingaretti. Complete classification of raw LIDAR data and 3D reconstruction of buildings. *Pattern Anal. Application*, 8:357–374, 2006.

R. Fowler. The thorny problem of LIDAR specifications. *Earth Observation Magazine*, 2001. URL `http://www.eomonline.com/Common/Archives/2001apr/01apr_fowler.html`.

P. Gamba. Digital surface models and building extraction: A comparison of IFSAR and LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 38(4):1959–1968, 2000.

M. Gerke, A. Koch, and B.-M. Straub. Building extraction from aerial imagery using a generic scene model and invariant geometric moments. In *Proceedings of the IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Area, IEEE Piscataway*, pages 85–89, 2001.

C. M. Gold. Triangulation based terrain modelling - where are we now? In R. Aangeenbury, editor, *Auto-Carto 4, International Symposium on Cartography and Computing*, pages 104 - 111, Baltimore, MD, 1979.

C. M. Gold. The Quad-Arc data structure. In T. Poiker and N. Chrisman, editors, *8th International Symposium on Spatial Data Handling*, pages 713–724, Vancouver, BC, Canada, 1998.

C. M. Gold and J. Snoeyink. A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30(2):144–163, 2001.

C. M. Gold, T. Charters, and J. Ramsden. Automated contour mapping using triangular element data. *Computer Graphics*, 11:170–175, 1977.

C. M. Gold, H. A. K. Charlesworth, and W. E. Kilby. Coal resource evaluation deformed sequences, using digital terrain models. *Bulletin of Canadian petroleum geology*, 29(2):259–266, 1981.

B. Gorte. Segmentation of TIN-structured surface models. In *Joint Conference on Geo-spatial theory, Processing and Applications, Ottawa*, page 5 pages, 2002.

L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

L. Guibas, D. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7:381–413, 1992.

N. Haala and C. Brenner. Extraction of buildings and trees in urban environments. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2-3):130–137, 1999.

M. Haklay. Map Calculus in GIS: A proposal and demonstration. *International Journal of Geographical Information Science*, 18(2):107–125, 2004.

O. Henricsson and E. Baltsavias. 3D building reconstruction with ARUBA: a qualitative and quantitative evaluation. In A. Grun, E. Baltsavias, and O. Henricsson, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II), Birkhauser, Basel*, pages 65–76, 1997.

A. D. Hofmann. Analysis of tin-structure parameter spaces in airborne laser scanner data for 3-d building model generation. In *XXth ISPRS Congress*, volume XXXV, Commission III, part B3, pages 302–307, Istanbul, Turkey, 2004. IAPRS.

A. D. Hofmann, H.-G. Maas, and A. Streilein. Derivation of roof types by cluster analysis in parameter spaces of airborne laserscanner point clouds. In *IAPRS International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, volume 34, Part 3/W13, pages 112–117, Dresden, Germany, 2003.

B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice.* Springer-Verlag Wien, 1992.

Y. Hu. *Automated Extraction of Digital Terrain Models, Roads and Buildings Using Airborne Lidar Data.* PhD thesis, University of Calgary, Department of Geomatics Engineering, October 2003.

K. Jacobsen and P. Lohmann. Segmented filtering of laser scanner DSMS. In A. S. G.V. H.-G. Maas, editor, *Proceedings of the ISPRS working group III/3 workshop '3-D reconstruction from airborne laserscanner and InSAR data'*, volume 34 Session IV, Dresden, Germany, 2003. Institute of Photogrammetry and Remote Sensing Dresden University of Technology.

X. Jiang and H. Bunke. Fast segmentation of range images into planar regions by scan line grouping. *Mach. Vis. Applicat.*, no. 7:115–122, 1994.

B. F. Kavanagh and S. J. G. Bird. *Surveying, Principles and Applications.* Prentice Hall, Upper Saddle River, New Jersey, 6th edition, 2002.

K. K. Kemp. Environmental modeling with GIS: A strategy for dealing with spatial continuity. In *Technical Report 93-3, National Center for Geographic Information and Analysis.* University of California, Santa Barbara, USA, 1993.

K. Khoshelham. Region refinement and parametric reconstruction of building roofs by integration of image and height data. In U. Stilla, F. Rottensteiner, and S. Hinz, editors, *CMRT05, IAPRS, Vol. XXXVI, Part 3/W24*, pages 3–8, Vienna, Austria, August 29-30 2005.

G. Limited. What is GPS?, 1996.

H.-G. Maas. Fast determination of parametric house models from dense airborne laserscanner data. In *ISPRS International Workshop on Mobile Mapping Technology*, volume 32, Part 2W1, pages 1–6, Bangkok, Thailand, 1999. International Archives of Photogrammetry and Remote Sensing.

H.-G. Maas and G. Vosselman. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54(2-3):153–163, 1999.

M. Mäntyla. *Methodological Background of the Geometric Workbench.* CAD-project. Helsinki University of Technology, Laboratory of Information Processing Science, Finland, 1981.

M. Mäntyla. Boolean operations of 2-manifolds through vertex neighborhood classification. *ACM Transactions on Graphics*, 5(1):1–29, 1986.

D. Maune. *Digital elevation model technologies and applications: the DEM users manual.* American Society of Photogrammetry and Remote Sensing, Bethesda, MD., 2001. URL `https://eserv.asprs.org/eseries/staticcontent/staticpages/1069.htm`.

E. M. Mikhail. Multi-source feature extraction and visualization in urban environments. In *International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences (IAPRS), Volume XXXIII, Workshop IC-13*, volume 33, pages 592–607, Amsterdam, 2000.

M. Morgan and A. Habib. Interpolation of LIDAR data and automatic building extraction. In *Proceedings of ACSM-ASPRS 2002 annual conference*, 2002.

M. Morgan and K. Tempeli. Automatic building extraction from airborne laser scanning data. In *Proceeding of the 19th ISPRS Congress, Book 3B, Amsterdam*, pages 616–623, 2000.

M. E. Mortenson. *Computer Graphics: An Introduction to the Mathematics and Geometry.* Industrial Press, New York, 1 edition, 1988.

S. Oude Elberink and H.-G. Maas. The use of anisotropic height texture measures for the segmentation of airborne laser scanner data. In *IAPRS. WG III/3., Vol XXXIII, Part B3.*, pages 678–684, Amsterdam, 2000.

N. Pfeifer and P. Stadler. Derivation od digital terrain models in the SCOP environment. In K. Torlegard, editor, *OEEPE workshop on airborne laserscanning and interfermoetric SAR for detailed digital elevation models*, Royal Institute of Technology, Departmen of Geodesy and Photogrammetry, 100 Stockholm, Sweden, 2001.

J. Rissanen. Universal coding, information, predicition, and estimation. *IEEE Transaction of Information Theory*, 30(4):629–636, 1984.

M. Roggero. Object segmentation with region growing and principal componenet analysis. In *ISPRS commission III Sysmposium "Photogrammetric Computer Vision"*, pages 298–294, Graz, Austria, 2002.

D. F. Rottensteiner. *Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS.* PhD thesis, Institute of Photogrammetry and Remote Sensing, Vienna University of Technology, Vienna, Austria, 2001.

F. Rottensteiner and C. Briese. A new method for building extraction in urban areas from high-resolution lidar data. In *International Archives of Photogrammetry and Remote Sensing, Symposium der ISPRS-Comm. III*, pages 295–301, Graz, Austria, 2002.

F. Rottensteiner and C. Briese. Automatic generation of building models from LIDAR data and the integration of aerial images. In H.-G. Maas, G. Vosselman, and A. Streilein, editors, *Proceedings of the ISPRS working group III/3 workshop '3-D reconstruction from airborne laserscanner and InSAR data'*, volume 34 Session IV, Dresden, Germany, 2003. Institute of Photogrammetry and Remote Sensing Dresden University of Technology.

F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik. Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis. *ISPRS Journal of Photogrammetry & Remote Sensing*, 62:135–149, 2007.

W. Shen. Building boundary extraction based on LIDAR point clouds data. *The international archives of the photogrammetry, remote sensing and spatial information sciences*, XXXVII, Part B3b:157–162, 2008.

G. Sithole. Filtering of laser altimetry data using a slope adaptive filter. In *IAPRS*, volume XXXIV - 3/W4, pages 203–210, Annapolis, MD, 2001.

G. Sithole and G. Vosselman. Report: ISPRS comparison of filters. Technical report, Department of Geodesy, Faculty of Civil Engineering and Geosciences, Delft University of Technology, The Netherlands, Delft, August 2003 2003.

G. Sohn and I. Dowman. Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. In *IAPRS, Vol XXXIV Part 3A. ISPRS Commission III, Symposium*, pages 336–344, Graz, Austria, 2002.

G. Sohn and I. Dowman. Extraction of buildings from high resolution satellite data. In *In: Baltsavias, E., Gruen, A., Van Gool, L. (Eds.), Automated Extraction of Man-Made Objects from Aerial and Space Images (III)*, pages 345–355. Balkema Publishers, Lisse, 2001.

G. Sohn and I. Dowman. Building extraction using lidar DEMS and IKONOS images. In H.-G. Maas, G. Vosselman, and A. Streilein, editors, *Proceedings of the ISPRS working group III/3 workshop '3-D reconstruction from airborne laserscanner and InSAR data'*, volume 34 Session IV, Dresden, Germany, 2003. Institute of Photogrammetry and Remote Sensing Dresden University of Technology.

G. Sohn and I. J. Dowman. Extraction of buildings from high resolution satellite data and LIDAR. In *ISPRS 20th Congress WGIII/4 Automated Object Extraction*, Istanbul, Turkey, 2004.

I. Suveg and G. Vosselman. 3d building reconstruction by map based generation and evaluation of hypotheses. In *BMVC01*, 2001.

I. Suveg and G. Vosselman. Reconstruction of 3D building models from aerial images and maps. *ISPRS Journal of Photogrammetry & Remote Sensing*, 58(3–4):202–224, 2004.

F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer. Hough-transform and extended RANSAC algorithms for automatic detection of 3D building roof planes from LIDAR data. *IAPRS ISPRS workshop on laser scanning*, XXXVI, Part 3/W52:407–412, 2007.

R. Tse and C. Gold. Terrain, dinosaurs and cadastres - options for three-dimension modelling. In C. Lemmen and P. van Oosterom, editors, *Proceedings: International Workshop on "3D Cadastres"*, pages 243–257, Delft, The Netherlands, 2001.

R. Tse and C. Gold. A proposed topological model for a three-dimensional cadastre. *Computer Environment and Urban Systems theme issue on '3D Cadastres'*, 27(4):427–445, 2003.

R. Tse and C. Gold. TIN meets CAD - extending the TIN concept in GIS. *Future Generation Computer Systems (Geocomputation)*, 20(7):1171–1184, 2004.

R. Tse and C. M. Gold. A surface-representation approach to a three-dimensional cadastre. *The GIM International*, 16(6):46–49, 2002a.

R. Tse, C. Gold, and D. Kidner. 3D urban terrain modelling with brdiges and holes. In *Proceedings, The 30th Congress of the International Geographical Union*, pages 15–20, Glasgow, UK, August 2004.

R. O. Tse. Semi-automated construction of fully three-dimensional terrain models. Master's thesis, The Hong Kong Polytechnic University, Hong Kong, May 2003.

R. O. Tse and C. M. Gold. Integration of terrain models and built-up structures using CAD-type Euler operators. In *Proceedings of International Workshop On Visualization and Animation Of Landscape*, Kunming, China, 2002b.

O. Veblen. Theory on plane curves in non-metrical analysis situs. *Transactions of the American Mathematical Society*, 6:pp. 83–98, 1905.

D. Veneziano, S. Hallmark, and R. Souleyrette. Accuracy of light detection and ranging derived terrain data for highway location. *Computer-Aided Civil and Infrastructure Engineering*, 19 (2):130–143, 2004.

G. Vosselman. Slope based filtering of laser altimetry data. In *IAPRS*, volume XXXIII, Part B3, pages 935–942, Amesterdam, The Netherlands, 2000.

G. Vosselman. 3d reconstruction of roads and trees for city modelling. In H.-G. Maas, G. Vosselman, and A. Streilein, editors, *Proceedings of the ISPRS working group III/3 workshop '3-D reconstruction from airborne laserscanner and InSAR data'*, volume 34, Part 3/W13, Dresden, Germany, 2003. Institute of Photogrammetry and Remote Sensing Dresden University of Technology.

G. Vosselman. Building reconstruction using planar faces in very high density height data. In *International Archives of Photogrammetry and Remote Sensing*, volume 32, part 3/2W5, pages 87–92, Munich, Germany, 1999a.

G. Vosselman. 3d measurements in images using CAD models. In *ASCT'99 conference*, Heijden, The Netherlands, 1999b.

G. Vosselman and S. Dijkman. 3D building model reconstruction from point clouds and ground plans. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 34, part 3/W4, pages 37–43, Annapolis, MA, USA, 2001.

G. Vosselman and H. Maas. Adjustment and filtering of raw laser altimetry data. In *OEEPE workshop on airborne laserscanning and interferometric SAR for detailed digital elevation models*, page paper 5(11 pages), Royal Institute of Technology, Department of Geodesy and Photogrammetry, 100 Stockholm, Sweden, 2001.

G. Vosselman, B. Gorte, G. Sithole, and T. Rabbani. Recognising structure in laser scanner point clouds. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 46, part 8/W2, pages 33–38, Freiburg, Germany, 2004.

R. Wack and A. Wimmer. Digital terrain models from airborne laser scanner data - a grid based approach. In *IAPRS, ISPRS Commission III, Symposium*, volume XXXIV, Part 3B, pages 293–296, Graz, Austria, 2002.

Z. Wang and T. Schenk. Building extraction and reconstruction from LIDAR data. *IAPRS, 17-22 July*, 33, part B3:958–964, 2000.

E. W. Weisstein. Normal vector. *From MathWord– A Wolfram Web Resource http://mathworld.wolfram.com/NormalVector.html*, 2007.

K. Zhang, J. Yan, and S.-C. Chen. Automatic construction of building footprints from airborne LIDAR data. *IEEE transactions on geoscience and remote sensing*, 44, No. 9:2523–2533, 2006.